REPORT DOCUMENTATION PAGE		Form Approved OMB No. 0704-0188

1. REPORT DATE ***(DD-MM-YYYY)***	**2. REPORT TYPE**	**3. DATES COVERED** *(From – To)*
07-05-2004	Final Report	7 June 2001 - 07-Jun-02

4. TITLE AND SUBTITLE Identification of Pseudo-random Sequences in DS/SS Intercepts by Higher-order Statistics	**5a. CONTRACT NUMBER** F61775-01-WE032
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Dr. Ernest R. Adams	**5d. PROJECT NUMBER**
	5d. TASK NUMBER
	5e. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Cranfield University (RMCS) Shrivenham SN6 8LA United Kingdom	**8. PERFORMING ORGANIZATION REPORT NUMBER** N/A
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) EOARD PSC 802 BOX 14 FPO 09499-0014	**10. SPONSOR/MONITOR'S ACRONYM(S)**
	11. SPONSOR/MONITOR'S REPORT NUMBER(S) SPC 01-4032

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

2004092O 086

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This report results from a contract tasking Cranfield University (RMCS) as follows: The contractor will investigate discriminant functions for known m-sequences and develop higher-order statistical methods to identify Gold codes in intercepts. The contractor will then assess the robustness of a detection to realistic channel effects that would occur for airborne RF signals.

15. SUBJECT TERMS

EOARD, Communications, Electromagnetics, Signal Processing

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18, NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			MICHAEL KJ MILLIGAN, Lt Col, USAF
UNCLAS	UNCLAS	UNCLAS	UL	64	**19b. TELEPHONE NUMBER** *(Include area code)* +44 (0)20 7514 4955

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39-18

AQ F04-11-1282

CONTRACT TITLE:	**Identification of Pseudo-random Sequences in DS/SS Intercepts by Higher-order Statistics.**
SPONSOR:	**Sensors Directorate (USAF), WPAFB, Dayton, Ohio, through EOARD, London.**
SPONSORS REFERENCE NUMBER:	**014032**
FINAL REPORT:	**Research carried out up to March 2004 by Dr E R Adams, Cranfield University, RMCS.**

1. Summary.

This report considers the use of higher-order statistics (HOS), in particular the triple-correlation function (TCF) and ninth-order correlation function (NCF), to detect the presence of m-sequences and Gold codes in signal intercepts. Section 2 describes the continuation of work in [1] to blindly detect m-sequences from the TCF. The new work reported in 2.4 improves detection rates significantly by searching for the TCF patterns associated with m-sequences likely to be present. Section 3 presents a new theory for the TCF of Gold codes. The expected TCF of Gold codes is 3-valued: no peaks of constituent m-sequences are present. The detection of a constituent m-sequence of a Gold code relies on the property that, at the locations of that m-sequence's TCF peaks, the Gold code TCF always has the expected value of $-1/L$, L the sequence length. Thus the Gold code TCF may be searched for sets of these values at known TCF peak locations for likely m-sequences. When a constituent m-sequence is found, a preferred-complementary m-sequence and the actual Gold code may also be easily found. Section 4 examines higher-order (>3) correlation functions. It is shown that Gold codes have NCF peaks. Although the NCF is very noisy, there are many peak locations, derivable from the TCF peak locations of the preferred pair of m-sequences, over which to average. The NCF method has the potential to produce good blind detection at high noise levels for long Gold codes. The final methods of section 5 were suggested by R Gold after a presentation by the author at AFRL, WPAFB in December 2002: a 1-stage triple-correlator detector for m-sequences and a 2-stage triple-correlator detector for Gold codes. Both detectors may be simply implemented with delay circuits, but their outputs are extremely noisy. They also provide the basis of phase-independent decoders.

With the exception of Dr Gold's suggestion and the simulation results for m-sequences in 2.3, produced by K Batty, all material reported is the original work of the author.

2. Detection and Identification of m-sequences.

It has been shown that m-sequences in signal intercepts may be detected and identified using HOS [1]. Subsequent work has demonstrated it is possible to automate the blind (unknown length and possible m-sequences present) derivation of an m-sequence's generator polynomial from a single TCF peak (see 2.1). Extensive simulation has revealed how intercept length and noise level influence detection rate (2.3). This extensive simulation was made possible by assuming hard decoding by the receiver: the chip values processed were +1 or −1 rather than real numbers, allowing fast binary arithmetic (2.2).

The simulation results show that the blind algorithms are inadequate for LPI signals, i.e. identification rates are low for SNRs below −5 dB, unless long or multiple intercepts are available. The present research is investigating the alternative strategy of searching for known m-sequences with known TCF peak locations (2.4).

2.1 Automatic blind m-sequence identification.

Firstly, the partial TCF $C'(p,q)$ of an N-length intercept x_i is calculated,

$$C'(p,q) = \frac{1}{N-q}\sum_{i=1}^{N-q} x_i x_{i+p} x_{i+q}, \quad 0 \le p \le q \le N-1,$$

$$C'(q,p) = C'(p,q) \quad \text{for} \quad 0 \le q < p \le N-1, \tag{2.1}$$

and peaks are detected. Each peak is used to estimate the length of an m-sequence present, and a majority vote taken to determine the final sequence length L. A single peak is then selected to derive the generator polynomial $g(X)$ of the m-sequence; other peaks may then be used to verify $g(X)$. A peak location (p,q) allows the algorithm to determine a unique $g(X)$ if the following criterion is satisfied:

$$\gcd(p,L) = 1 \ \vee\ \gcd(q,L) = 1, \tag{2.2}$$

where gcd is greatest common divisor and $\vee$ is logical 'or'. Note, when L is prime, all (p,q) determine a $g(X)$.

Two algorithms have been developed [2] to derive $g(X)$ from a (p,q) satisfying (2.2).

(i) Gauss elimination over GF(2) for the m equations:

$$\begin{aligned} &\alpha^{p} + \alpha^{q} = 1, \\ &\alpha^{2p \bmod L} + \alpha^{2q \bmod L} = 1, \\ &\ldots \\ &\alpha^{2^{m-1}p \bmod L} + \alpha^{2^{m-1}q \bmod L} = 1. \end{aligned} \tag{2.3}$$

(ii) Use the derived equation:

$$\alpha^{p} + \alpha^{q} = 1 \Rightarrow \alpha^{p-q} + \alpha^{-q} = 1 \tag{2.4}$$

to rewrite any α^n, in equations derived from peaks, in terms of reduced powers of α:

$$\alpha^{n} = \alpha^{n-q}\left(\alpha^{p} + 1\right). \tag{2.5}$$

Fault tolerant software, using either (i) or (ii), has been written which has produced a $g(X)$ in all cases tried.

More details of the above methods may be found in [2,3].

2.2 Noise assumptions for hard-decision receivers.

HOS suppress the influence of Gaussian noise signals, their cumulants above order two vanishing. A partial TCF is a third-order moment which closely approximates the third-order cumulant as the expected average value of a partial m-sequence is $-1/L$, close to zero for large L. For processes with zero mean, third-order moments and cumulants are identical.

The statistical properties of partial TCF estimators are derived in [1] for m-sequences whose values are corrupted by AWGN. For N(0,σ^2) noise, the additional variance of $C'(p,q)$ is

$$\mathrm{var}_{\sigma^2} = \left(\frac{\sigma^6 + 3\sigma^4 + 3\sigma^2}{N-q} \right) \quad \text{for} \quad p \neq q. \tag{2.6}$$

An alternative noise model results if the receiver makes a hard decision for each chip, assuming the same AWGN. Sufficient noise may result in a transmitted 1 being received as −1 or vice versa. Assuming a threshold of zero for the hard decision, the probability of mistaking 1 for −1, or −1 for 1, is

$$\beta = \frac{1}{\sigma\sqrt{2\pi}} \int_1^{\infty} e^{-x^2/2\sigma^2}\,\mathrm{d}x. \tag{2.7}$$

Thus the 3-valued chip error y_i has the probability distribution:

$$\begin{aligned} P(y_i = 0) &= 1 - 2\beta, \\ P(y_i = 2) &= \beta, \\ P(y_i = -2) &= \beta. \end{aligned} \tag{2.8}$$

The partial TCF of $x_i = u_i + y_i$, for m-sequence sample u_i with superimposed error sequence y_i , is

$$\begin{aligned} C'_{u,y}(p,q) &= \frac{1}{N-q} \sum_{i=1}^{N-q} (u_i + y_i)(u_{i+p} + y_{i+p})(u_{i+q} + y_{i+q}) \\ &= C'_u(p,q) + C'_y(p,q) + \sum_{j=1}^{6} T_j \\ T_j &= \frac{1}{N-q} \sum_{i=1}^{N-q} \tau_{ji}. \end{aligned} \tag{2.9}$$

Cases $p = q$, $p = 0$ and $q = 0$ are not considered, as there are no peaks for such values.

$C'_u(p,q)$: the partial TCF of u is unbiased, with variance

$$\mathrm{var}[C'_u(p,q)] = \begin{cases} \dfrac{1}{N-q} & \text{for non peak } (p,q) \\ 0 & \text{for peak } (p,q) \end{cases} \tag{2.10}$$

$C'_y(p,q)$:

$$\mathrm{E}[y_i y_{i+p} y_{i+q}] = 0, \qquad \mathrm{var}[y_i] = 8\beta. \tag{2.11}$$

Thus

$$\mathrm{E}[C'_y(p,q)] = 0, \qquad \mathrm{var}[C'_y(p,q)] = \frac{512\beta^3}{N-q}. \tag{2.12}$$

T_1, T_2, T_3:

$$\tau_{1i} = u_{i+k}\, y_{i+q} \qquad \text{for} \qquad u_{i+k} = u_i u_{i+p} \quad ,$$

$$\tau_{2i} = u_{i+l}\, y_{i+p} \qquad \text{for} \qquad u_{i+l} = u_i u_{i+q} \quad ,$$

$$\tau_{3i} = u_{i+s}\, y_i \qquad \text{for} \qquad u_{i+s} = u_{i+p}\, u_{i+q}. \tag{2.13}$$

T_1, T_2 and T_3 are nearly orthogonal combinations of y values:

$$\begin{aligned} &\mathrm{E}[\tau_{1i}] = 0 \\ &\mathrm{var}[\tau_{1i}] = \mathrm{E}\left[u_{i+k}^2\, y_{i+q}^2\right] = 8\beta \end{aligned} \tag{2.14}$$

Clearly τ_{2i} and τ_{3i} have the same mean and variance, and

$$\left.\begin{aligned} &\mathrm{E}[T_i] = 0 \\ &\mathrm{var}[T_i] = \frac{8\beta}{N-q} \end{aligned}\right\} 1 \le i \le 3 \tag{2.15}$$

T_4, T_5, T_6:

$$\tau_{4i} = u_i\, y_{i+p}\, y_{i+q} \quad ,$$

$$\tau_{5i} = u_{i+p}\, y_i\, y_{i+q} \quad ,$$

$$\tau_{6i} = u_{i+q}\, y_i\, y_{i+p} \quad . \tag{2.16}$$

Again, the T_j are nearly orthogonal combinations of the products of independent y values:

$$\begin{aligned} &\mathrm{E}[\tau_{4i}] = 0\,, \\ &\mathrm{var}[\tau_{4i}] = \mathrm{E}\left[u_i^2\, y_{i+p}^2\, y_{i+q}^2\right] = 64\beta^2\,, \end{aligned} \tag{2.17}$$

so

$$\begin{aligned} &\mathrm{E}[T_4] = \mathrm{E}[T_5] = \mathrm{E}[T_6] = 0\,, \\ &\mathrm{var}[T_4] = \mathrm{var}[T_5] = \mathrm{var}[T_6] = \frac{64\beta^2}{N-q}\,. \end{aligned} \tag{2.18}$$

$C'_{u,y}$ is therefore an unbiased estimator for u's TCF for $p \neq q$, $p \neq 0$ and $q \neq 0$. As the above contributions are nearly independent, the overall variance for non-peak (p,q) is approximately:

$$\mathrm{var}\left[C'_{u,y}(p,q)\right] = \frac{512\beta^3 + 192\beta^2 + 24\beta + 1}{N-q} \tag{2.19}$$

For peak (p,q) this variance (2.19) is reduced by $1/(N-q)$. These approximate variances were used in algorithms to detect peaks and determine sequence length L.

TCF computations are significantly faster when noise is represented as chip errors. Multiplications may be replaced with XORs of the binary data in the C programs.

2.3 **Simulation results for blind identification.**

A set of simulations was conducted to investigate the joint effects of probability of chip error P(e) and intercept length N on identification rate. The m-sequences used were of length L = 127, knowledge denied to the algorithm. A series of 1000 trials was carried out for a set of P(e) varying from 0.01 to 0.2 and N varying from 60 to 120. Results are shown in Figure 2.1.

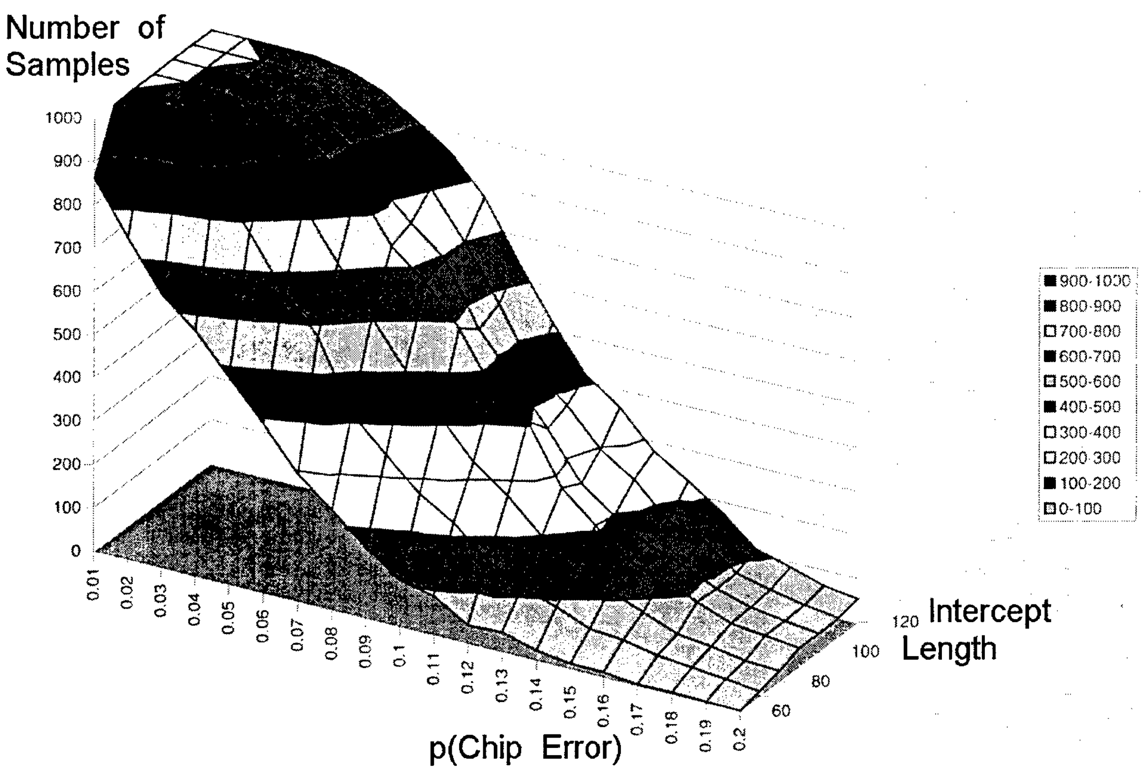


Figure 2.1 Influence of chip errors and intercept length on blind $g(X)$ identification.

Note that P(e) = 0.06 corresponds with a SNR of approximately 3 dB. For 3 dB: when N = 120 the identification rate is 96%; when intercept length is halved to N = 60, the

identification rate is almost halved to 51%. Even for large N, close to L, identification rates are poor for large P(e), or low SNR. They may only be improved by using multiple intercepts.

2.4 Pattern recognition to identify known m-sequences

The number of primitive polynomials S_m of order m, each generating an m-sequence, satisfies:

$$S_m = \frac{2^m - 2}{m} \quad \text{when} \quad L = 2^m - 1 \text{ is prime,}$$

$$S_m < \frac{2^m}{m} \quad \text{for non - prime } L. \tag{2.20}$$

Very few of the TCF peaks of the generated m-sequences are common with those of m-sequences of the same or different length.

The total number of primitive polynomials up to order 13, i.e. m-sequences up to length 8,191, is 1,108:

$$\sum_{m}^{13} S_m = 1{,}108. \tag{2.21}$$

This is not a large search space. The number of primitive polynomials up to order of 17 is 7,710; the number up to order 19 is 27,594.

Discriminant functions of the form

$$D_u = \sum_{i=1}^{n_u} \lambda_i C'(p_i, q_i) \tag{2.22}$$

were derived for a set of m-sequences of length 1,023. Each of the n_u peaks (p_i,q_i) are unique to sequence u. Peaks close to the origin were chosen so they could be evaluated from short samples. The weightings λ_i (2.22) of TCF values at peak locations i reflect the variance of those values {approximately $1/(N - q_i)$} and modulation effects.

Simulations to estimate identification rate were carried out for noisy m-sequences and noisy modulated m-sequences. For unmodulated sequences, the λ_i were inversely proportional to variances:

$$\lambda_i = N - q_i \quad , \quad q_i > p_i . \tag{2.23}$$

For randomly modulated sequences, the λ_i were estimated by multivariate analysis. In each run, the sequence was identified when the appropriate D_u (2.22) exceeded a threshold. Two intercept lengths were used: 127, when the D_u were linear combinations of 5 peaks ($n_u = 5$); 255, when 20 peaks were used. The results, summarized in tables 2.1 and 2.2, are significantly better than those for blind identification in section 2.3.

		SAMPLE / 1023	
		127 (5 peaks)	255 (20 peaks)
SNR	3 dB	96.7 (0.5)	100 (0)
	0 dB	88.3 (1.3)	99.3 (0)

Table 2.1 Identification rates for m-sequences with no modulation.

		SAMPLE / 1023	
		127 (5 peaks)	255 (20 peaks)
SNR	3 dB	84.5 (0.7)	99.5 (0)
	0 dB	76.1 (1.2)	97.4 (0)

Table 2.2 Identification rates for randomly modulated m-sequences.

For the tabulated $x(y)$ values, x is the % identification of an m-sequence and y the % false alarms, the detection of a sequence not present.

3. Triple-Correlation Based Detection and Identification of Gold Codes

Gold codes are more difficult to detect than m-sequences from their triple-correlation function. They are not finite-field elements and do not obey the shift-and-multiply rule. There are no triple-correlation peaks except when sequence length L is divisible by 3, when the two peaks ($L/3$, $2L/3$) and ($2L/3$, $L/3$) on the diagonal $p + q = L$ are shared by all L-length sequences, e.g. for $L = 15$, there are peaks at (5,10) and (10,5).

Gold codes have a looser structure than m-sequences: shift-and-multiplying produces not another cyclically shifted version of the same code but another code in the same family, i.e. one generated by the same two preferred m-sequences but with a different relative phase (see 3.2). It will be shown that the triple-correlation function of Gold codes contains sufficient information for the detection of constituent m-sequences and, by further correlation processing, the actual Gold code.

3.1. Preferred m-sequences.

Gold codes are formed by modulo-2 adding the bits of a preferred pair of m-sequences. Preferred pairs have the defining property that their cross-correlation function is 3-valued. For a preferred pair of m-sequences of length $L = 2^n - 1$, these three values are [4]:

$$\theta_{-1} = -\mathrm{t}(n)/L$$

$$\theta_0 = -1/L$$

$$\theta_1 = \{\mathrm{t}(n) - 2\}/L \qquad (3.1)$$

where

$$t(n) = \begin{cases} 1+2^{(n+1)/2} \text{ for odd } n \\ 1+2^{(n+2)/2} \text{ for even } n \end{cases} \tag{3.2}$$

Note that there are no preferred pairs for $n = 0 \bmod 4$.

Preferred pairs may be constructed by decimation. For sequences $u = (u_0, u_1, .., u_{L-1})$ and $v = (v_0, v_1, .., v_{L-1})$,

$$v = u[q] \Rightarrow \forall i (0 \le i \le L-1)\ v_i = u_{qi \bmod L}. \tag{3.3}$$

$u[q]$ is a proper decimation of u, i.e. has period L, iff gcd $(L, q) = 1$.

u and v are a preferred pair if $n \neq 0 \bmod 4$ and $v = u[q]$, where q is odd and

$$q = 2^k + 1 \text{ or } q = 2^{2k} - 2^k + 1,.$$

$$\gcd(n, k) = \begin{cases} 1 \text{ for odd } n \\ 2 \text{ for } n = 2 \bmod 4 \end{cases} \tag{3.4}$$

For $L = 31$ ($n = 5$), the preferred pairs of m-sequences are those connected by a straight line in Figure 3.1[4].

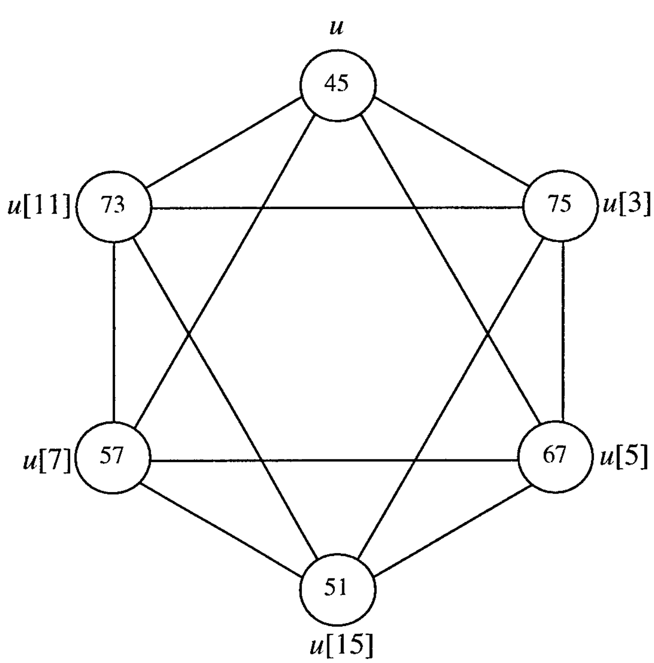


Figure 3.1 Preferred pairs of m-sequences for $L = 31$.

For u, 45 is the octal form of its generator polynomial's coefficients:

$$g_u(X) = X^5 + X^2 + 1 \tag{3.5}$$

3.2. Derivation of Gold codes from preferred pairs of m-sequences.

Gold codes may be generated from a pair of preferred sequences u and v by modulo-2 adding u and cyclically shifted versions of v, or vice versa. If T denotes the operator which cyclically shifts vectors one place to the left:

$$Tv = (v_1, v_2, .., v_{L-1}, v_0),$$

$$T^k v = (v_k, v_{k+1}, .., v_{L-1}, v_0, .., v_{k-1}) \text{ for } 0 \le k < L \tag{3.6}$$

then the following set of L+2 Gold codes results:

$$G(u,v) = \{u, v, u \oplus v, u \oplus Tv, u \oplus T^2 v, .., u \oplus T^{L-1} v\} \tag{3.7}$$

where $\oplus$ denotes modulo-2 addition of corresponding sequence bits. For the L Gold sequences

$$w^k = u \oplus T^k v, \quad 0 \le k \le L-1 \tag{3.8}$$

individual bits may be expressed:

$$w_i^k = u_i \oplus v_{(i+k) \bmod L}, \quad 0 \le i \le L-1 \tag{3.9}$$

As an example, the preferred pair of sequences u and $v = u[3]$ in Figure 3.1, octal 45 and 75, and $w^0 = u \oplus v$ are:

$$\begin{aligned} u &= 1\,0\,1\,0\,1\,1\,1\,0\,1\,1\,0\,0\,0\,1\,1\,1\,1\,1\,0\,0\,1\,1\,0\,1\,0\,0\,1\,0\,0\,0\,0 \\ v = u[3] &= 1\,0\,1\,1\,0\,1\,0\,1\,0\,0\,0\,1\,1\,1\,0\,1\,1\,1\,1\,1\,0\,0\,1\,0\,0\,1\,1\,0\,0\,0\,0 \\ w^0 &= 0\,0\,0\,1\,1\,0\,1\,1\,1\,1\,0\,1\,1\,0\,1\,0\,0\,0\,1\,1\,1\,1\,1\,1\,0\,1\,0\,0\,0\,0\,0 \end{aligned} \tag{3.10}$$

Note that, if bits are mapped $\{0 \mapsto 1, 1 \mapsto -1\}$, the ± 1 values are multiplied:

$$w_i^k = u_i v_{(i+k) \bmod L}, \quad 0 \le i \le L-1 \tag{3.11}$$

e.g.

$$w^o = 111{-1}{-1}1{-1}{-1}{-1}{-1}1{-1}{-1}1{-1}111{-1}{-1}{-1}{-1}{-1}{-1}1{-1}111111 \tag{3.12}$$

3.3. Autocorrelation functions (ACFs) of Gold sequences.

For the {1,-1} representation, assuming all indices are mod L, the ACF of w^k is given by

$$C^k(p)=\frac{1}{L}\sum_{i=0}^{L-1} w_i^k w_{i+p}^k$$

$$=\frac{1}{L}\sum_{i=0}^{L-1} u_i v_{i+k} u_{i+p} v_{i+k+p} \tag{3.13}$$

$$=\frac{1}{L}\sum_{i=0}^{L-1} u_{i+r_p} v_{i+s_{kp}}$$

for some $r_p, s_{kp} \in \{0,1,..,L-1\}$, r_p determined by p and s_{kp} by k and p. Thus $C^k(p)$ reduces to the cross-correlation between two preferred m-sequences, and has the values:

$$C^k(0)=1$$

$$C^k(p)=\theta_{-1}, \theta_0 \; or \; \theta_1, \text{ as given in (3.1), for } 1 \le p \le L-1 \tag{3.14}$$

Partial correlation functions of Gold sequences may take a wider range of values, which depend on initial position in the sequence. Thus the following partial ACF of w^k, calculated from an N-sum ($N \le L$), depends on j and p:

$$C_N^k(j,p)=\frac{1}{N}\sum_{i=j}^{N+j-1} w_i^k w_{i+p}^k \tag{3.15}$$

But, using group closure (shift-and-multiply rule) for m-sequences u and v:

$$w_i^k w_{i+p}^k = u_i v_{i+k} u_{i+p} v_{i+k+p}$$

$$= u_i u_{i+p} v_{i+k} v_{i+k+p} \tag{3.16}$$

$$= u_{i+r_p} v_{i+s_{kp}}, \; r_p \text{ and } s_{kp} \in \{0,1,..,L-1\}$$

$$= w_{i+r_p}^{\ell}, \; \ell = s_{kp} - r_p$$

Thus

$$C_N^k(j,0)=1$$

$$C_N^k(j,p)=\frac{1}{N}\sum_{i=j}^{N+j-1} w_{i+r_p}^{\ell}, \; 1 \le p \le L-1 \tag{3.17}$$

The partial correlation of w^k reduces to the mean of a partial sequence within another w^{ℓ} from *G(u,v)*.

3.4. Triple-Correlation Functions (TCFs) of Gold Sequences.

3.4.1 TCFs of Complete Gold Sequences

The TCF of w^k is defined:

$$C^k(p,q) = \frac{1}{L}\sum_{i=0}^{L-1} w_i^k w_{i+p}^k w_{i+q}^k \tag{3.18}$$

Substituting from (3.16)

$$C^k(p,q) = \frac{1}{L}\sum_{i=0}^{L-1} w_{i+r_p}^{\ell} w_{i+q}^k \tag{3.19}$$

The TCF is thus the cross-correlation of two Gold sequences from $G(u,v)$. This cross-correlation is 3-valued [4], with the values given in (3.1), so

$$C^k(p,q) = \theta_{-1}, \theta_0 \ or \, \theta_1 \tag{3.20}$$

(3.18) may be rewritten:

$$C^k(p,q) = \frac{1}{L}\sum_{i=0}^{L-1} u_i v_{i+k} u_{i+p} v_{i+k+p} u_{i+q} v_{i+k+q} \tag{3.21}$$

Consider the case where (p',q') is a TCF peak location for m-sequence u, i.e. $u_i = u_{i+p'} u_{i+q'}$ for $0 \le i \le L-1$. As $u_i u_{i+p'} u_{i+q'} = u_i^2 = 1$:

$$C^k(p',q') = \frac{1}{L}\sum_{i=0}^{L-1} v_{i+k} v_{i+k+p'} v_{i+k+q'} \tag{3.22}$$

For prime L and nearly all other cases when L is not prime [1], if (p',q') is a TCF peak for u it cannot be a TCF peak for v, i.e. it is impossible that

$$v_{i+k} = v_{i+k+p'} v_{i+k+q'} \ \text{for all } i \, (0 \le i \le L-1) \tag{3.23}$$

Closure (shift-and-multiply rule) gives

$$v_{i+k+p'} \, v_{i+k+q'} = v_{i+k+r_{p'q'}}, \ r_{p'q'} \ne 0 \tag{3.24}$$

for some $r_{p'q'}$ $(1 \le r_{p'q'} \le L-1)$ determined by (p',q').

Thus, as phase is irrelevant to the TCF:

$$C^k(p',q') = \frac{1}{L}\sum_{i=0}^{L-1} v_{i+k} v_{i+k+r_{p'q'}}$$

$$= \frac{1}{L}\sum_{i=0}^{L-1} v_i v_{i+r_{p'q'}} \tag{3.25}$$

But this expression for the ACF of v yields just one value as $r_{p'q'}$ cannot be zero:

$$r_{p'q'} \neq 0 \Rightarrow C^k(p',q') = \frac{-1}{L} \tag{3.26}$$

A similar reduction occurs when (p'',q'') is a TCF peak for v, and automatically not for u:

$$C^k(p'',q'') = \frac{1}{L}\sum_{i=0}^{L-1} u_i u_{i+p''} u_{i+q''}$$

$$= \frac{1}{L}\sum_{i=0}^{L-1} u_i u_{i+s_{p''q''}},\ s_{p''q''} \neq 0 \tag{3.27}$$

$$= \frac{-1}{L}$$

Note that, from (3.21) when $p=q$:

$$C^k(p,p) = \frac{1}{L}\sum_{i=0}^{L-1} u_i v_{i+k} = \frac{1}{L}\sum_{i=0}^{L-1} w_i^k \tag{3.28}$$

for all p $(0 \leq p \leq L-1)$. This, the cross-correlation of two preferred m-sequences, is 3-valued: θ_{-1}, θ_0 or θ_1 are the possible values for different k.

Also, as, for all α, $v_\alpha^2 = u_\alpha^2 = 1$,

$$C^k(0,q) = \frac{1}{L}\sum_{i=0}^{L-1} u_{i+q} v_{i+k+q}$$

$$C^k(p,0) = \frac{1}{L}\sum_{i=0}^{L-1} u_{i+p} v_{i+k+p} \tag{3.29}$$

$$C^k(0,0) = \frac{1}{L}\sum_{i=0}^{L-1} u_i v_{i+k}$$

But these three values are equal:

$$C^k(0,q)=C^k(p,0)=C^k(0,0)=C_{(u,v)}(k) \tag{3.30}$$

where $C_{(u,v)}(k)$ is the cross-correlation between u and v for shift k, known to be 3-valued.

Thus, when $p = q$ or either or both of p and q are zero, the same value θ_{-1}, θ_0 or θ_1 results, depending on k:

$$p=q \vee p=0 \vee q=0 \Rightarrow C^k(p,q)=C_{(u,v)}(k)=\frac{1}{L}\sum_{i=0}^{L-1} w_i^k \tag{3.31}$$

The $L-1$ (p',q') and $L-1$ (p'',q'') values resulting in TCF peaks for exclusively u or v produce $2L$-2 values of θ_0 for the TCF of w^k:

$$\{(p,q)=(p_i',q_i') \oplus (p,q)=(p_i'',q_i'')\},\ 1\le i\le L-1 \Rightarrow C^k(p,q)= \theta_0 =\frac{-1}{L} \tag{3.32}$$

where $\oplus$ denotes 'exclusive or'. For all other $C^k(p,q)$, which may be regarded as a cross-correlation function of two Gold sequences from $G(u,v)$:

$$(p,q)\neq(p_i',q_i') \wedge (p,q)\neq(p_i'',q_i'') \wedge p\neq q \Rightarrow C^k(p,q)=\theta_{-1},\theta_0 \text{ or } \theta_1. \tag{3.33}$$

If there are equal numbers of (p,q) values producing θ_{-1} and θ_1, the mean of the resulting $C^k(p,q)$ is

$$\frac{(\theta_{-1}+\theta_1)}{2}=\frac{-t(n)+t(n)-2}{2L}=\frac{-1}{L} \tag{3.34}$$

$$C^k(p,q) = \frac{1}{L}\sum_{i=0}^{L-1} w_i^k w_{i+p}^k w_{i+q}^k$$

Equivalent forms

$$\frac{1}{L}\sum_i w_{i+r}^l w_{i+q}^k \quad = \quad \frac{1}{L}\sum_i u_i v_{i+k} u_{i+p} v_{i+k+p} u_{i+q} v_{i+k+q}$$

θ_{-1}, θ_0 or θ_1

$(p,q) = (p',q')$

a TCF peak location for u,

i.e. $u_i u_{i+p'} u_{i+q'} = 1$.

$$\frac{1}{L}\sum_i v_{i+k} v_{i+k+p'} v_{i+k+q'}$$

(p',q') a peak for u

$\Rightarrow$ NOT a peak for v

$$\frac{1}{L}\sum_j v_j v_{j+r} \left(r \neq 0\right)$$

$$\theta_0 = \frac{-1}{L}$$

Figure 3.1 Alternative TCF Values

3.4.2 TCFs of Partial Gold Sequences

The partial TCF of w^k may be defined $(N \leq L)$:

$$C_N^k(j,p,q) = \frac{1}{N}\sum_{i=j}^{N+j-1} w_i^k w_{i+p}^k w_{i+q}^k \ , \tag{3.35}$$

which may again be simplified using (3.16):

$$C_N^k(j,p,q) = \frac{1}{N}\sum_{i=j}^{N+j-1} w_{i+r_p}^{\ell} w_{i+q}^k \tag{3.36}$$

which is the partial cross-correlation between two subsequences of different Gold sequences from $G(u,v)$. (3.36) may be rewritten:

$$C_N^k(j,p,q) = \frac{1}{N}\sum_{i=j}^{N+j-1} u_i v_{i+k} u_{i+p} v_{i+k+p} u_{i+q} v_{i+k+q} \tag{3.37}$$

Rewriting (3.37) for the general case, where (p,q) is not necessarily a peak for u or v:

$$\begin{aligned} C_N^k(j,p,q) &= \frac{1}{N}\sum_{i=j}^{N+j-1} u_i u_{i+p} u_{i+q} v_{i+k} v_{i+k+p} v_{i+k+q} \\ &= \frac{1}{N}\sum_{i=j}^{N+j-1} u_{i+\alpha_{pq}} v_{i+\beta_{kpq}} \qquad (3.38) \\ &= \frac{1}{N}\sum_{i=\lambda}^{N+\lambda-1} w_i^m \ , \ \lambda = j + \alpha_{pq} \ \text{ and } \ m = \beta_{kpq} - \alpha_{pq} \end{aligned}$$

Thus the general TCF may be regarded as a cross-correlation between two Gold sequences, a cross-correlation of a pair of preferred m-sequences, or the average value of a Gold sequence. For a complete sequence ($N = L$), all these are 3-valued.

For (p',q') corresponding to a TCF peak for u and not for v:

$$\begin{aligned} C_N^k(j,p',q') &= \frac{1}{N}\sum_{i=j}^{N+j-1} v_{i+k} v_{i+k+p'} v_{i+k+q'} \\ &= \frac{1}{N}\sum_{i=j+k}^{N+j+k-1} v_i v_{i+p'} v_{i+q'} \qquad (3.39) \\ &= \frac{1}{N}\sum_{i=j+k}^{N+j+k-1} v_i v_{i+r_{p'q'}} \ , \ r_{p'q'} \neq 0 \end{aligned}$$

Thus $C_N^k(j,p',q')$ is a partial ACF value (for non-zero shift) of an m-sequence, with a mean of $-1/L$, and its statistical properties independent of k and j. Similar values arise if (p,q) is a peak for v and not for u.

Again, if $p=q$ or either or both of p and q are zero, the average value of the partial TCF is θ_{-1}, θ_0 or θ_1, depending on k.

3.5 Statistical theory of Gold sequence TCFs.

For (p,q) exclusively a peak for u or v, the partial TCF in (3.39) reduces to a partial ACF of an m-sequence. [1] gives the following mean and variance:

$$\mathrm{E}\left[C_N^k(j,p,q)\right]=\frac{-1}{L}$$

$$\mathrm{var}\left[C_N^k(j,p,q)\right]=\frac{1}{N'}\left\{1-\frac{N'-1}{L}\right\}-\frac{1}{L^2} \tag{3.40}$$

As an N-length intercept is available, $N'=N-q$ products are used to estimate $C_N^k(j,p,q)$, assuming $p\leq q$.

For the more general partial TCF expression (3.38) when (p,q) is not necessarily a u or v peak, there are three possible cases: the mean value (actual value when $N=L$) may be θ_{-1}, θ_0 or θ_1. Assuming each chip of the Gold sequence may be regarded as independent of the other chips, the first two moments about zero of the partial TCF are:

$$\mathrm{E}\left[C_N^k(j,p,q)\right]=\frac{1}{N}\mathrm{E}\left[\sum_{i=\lambda}^{N+\lambda-1}w_i^m\right]$$

$$=\frac{1}{N}\theta_\ell N=\theta_\ell \tag{3.41}$$

$$\mathrm{E}\left[\left\{C_N^k(j,p,q)\right\}^2\right]=\sum_{i=\lambda}^{N+\lambda-1}\sum_{j=\lambda}^{N+\lambda-1}\mathrm{E}\left[w_i^m w_j^m\right]$$

$$=\frac{1}{N^2}\left\{N+\left(N^2-N\right)\delta_\ell\right\}=\frac{1}{N}\left\{1+(N-1)\delta_\ell\right\} \tag{3.42}$$

where

$$\delta_\ell=\mathrm{E}\left[w_i^m w_j^m\right]\ ,\ i\neq j. \tag{3.43}$$

So the variance is

$$\operatorname{var}\left[C_N^k(j,p,q)\right]=\frac{1}{N}\left\{1+(N-1)\delta_\ell-\theta_\ell^2 N\right\} \tag{3.44}$$

Because this variance is zero when $N = L$ (the partial TCF becomes θ_ℓ), setting (3.44) to zero gives

$$\delta_\ell=\frac{\theta_\ell^2 L-1}{L-1} \tag{3.45}$$

Substituting (3.45) into (3.44) gives

$$\begin{aligned}\operatorname{var}\left[C_N^k(j,p,q)\right]&=\frac{1}{N}\left\{1-\frac{N-1}{L-1}-\frac{(L-N)\,\theta_\ell^2}{L-1}\right\}=\frac{(L-N)\left(1-\theta_\ell^2\right)}{N(L-1)}\\&\cong\frac{1}{N}\left(1-\frac{N-1}{L}\right)\end{aligned} \tag{3.46}$$

assuming L is large and θ_ℓ^2 is small. The variance approximation in (3.46) is valid in all three above cases. The Central Limit Theorem may be used to show all three populations are approximately Gaussian with the same variance but means of θ_{-1}, θ_0 and θ_1.

For the case $\ell=0$ $(\theta_0=-1/L)$ the exact variance is:

$$\operatorname{var}\left[C_N^k(j,p,q)\right]=\frac{1}{N}\left\{1-\frac{N-1}{L-1}-\frac{L-N}{L^2(L-1)}\right\} \tag{3.47}$$

The expression in (3.47) is asymptotically identical to the variance in (3.40), interpreting N' and N as the number of products used to estimate the partial TCF.

3.6 Statistical tests for constituent m-sequences of a Gold code.

In the ideal case of a noiseless L-length intercept of a Gold code, the presence of constituent m-sequences u and v would be indicated by the $-1/L$ TCF values at all peak locations (p_i', q_i') for u and (p_i'', q_i'') for v. For an N-length intercept ($N<L$), the mean TCF values at peak locations is still $-1/L$, but with the approximate variance σ_0^2 given in (3.46). However, for a set of TCF values from non-peak locations, the mean value would also be $-1/L$, as the expected numbers with average values of θ_{-1} and θ_1 are equal, and their average is $-1/L$ (3.34). The approximate variance for such a set is given by

$$\sigma_{\pm 1}^2 \cong \sigma_0^2 + \gamma\left(\frac{\mathrm{t}(n)-1}{L}\right)^2 = \sigma_0^2 + \gamma\left\{\frac{2^{\lfloor (n+2)/2 \rfloor}}{L}\right\}^2 = \sigma_0^2 + d^2\,, \tag{3.48}$$

where $\lfloor x \rfloor$ denotes the largest integer not exceeding x and γ is the proportion of TCF values taking the values $-t(n)/L$ *or* $\{t(n)-2\}/L$. The theoretical value for γ is ½. This approximation (3.48) is an *under*estimate, but is used for convenience in the tests below, which therefore yield pessimistic detection rates. The actual variance, assuming σ_0^2 is exact, is

$$\sigma_{\pm 1}^2 = \frac{1}{\sigma_0\sqrt{2\pi}} \int_{-\infty}^{+\infty} x^2 \mathrm{e}^{-(x-d)^2/2\sigma_0^2}\,\mathrm{d}x \tag{3.49}$$

It is proposed that tests for u and v be based on the different *variances* of sets of TCF values at peak and at non-peak locations. For any L, it is easy to tabulate peak locations for all m-sequences. The majority of locations are non-peak and a set of these may be chosen as a control set. An approximate test may be carried out on whether the variance of the control set is significantly greater than the variance of the proposed set of peaks by calculating the ratio of the variances and comparing with an F-statistic. The following sum of squared deviations from $-1/L$ is computed:

$$Z = \sum_{i=1}^{\gamma}\left\{C_N^k(j, p_i, q_i) + \frac{1}{L}\right\}^2 \tag{3.50}$$

For Z_u the value of Z when (p_i,q_i) are γ_u peaks for u, and Z_c when the γ_c peaks (p_i,q_i) are the non-peaks in the control set, the variance-ratio may be compared with tabulated F values:

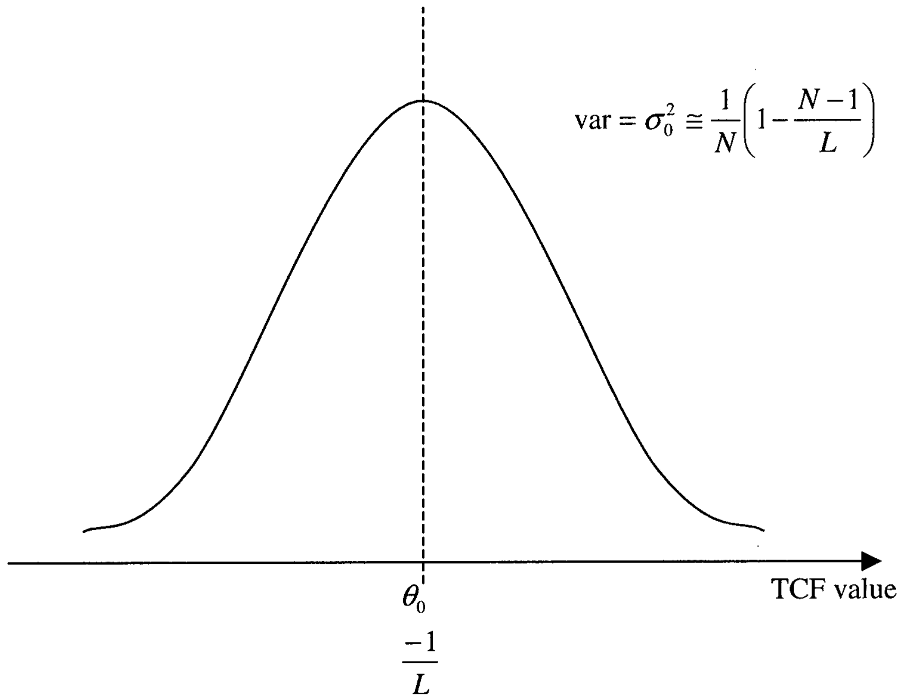


Figure 3.2 PDF of set of TCF values including just peak locations.

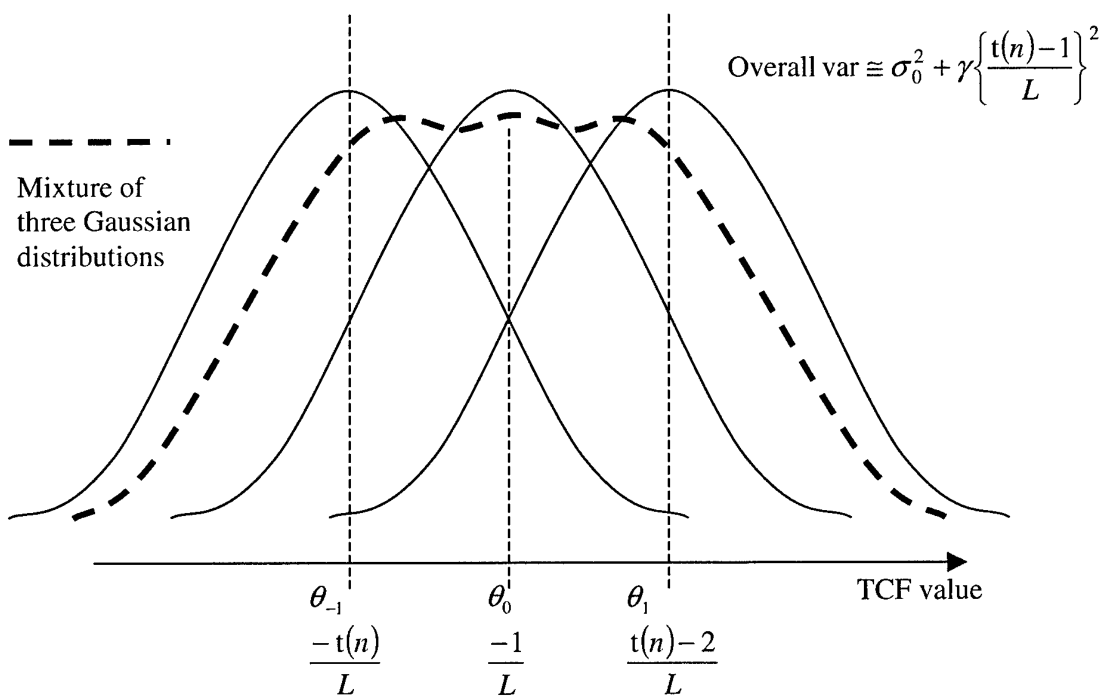


Figure 3.3 PDF of control set of TCF values including no peaks.

$$\frac{Z_c/(\gamma_c-1)}{Z_u/(\gamma_u-1)} \geq \mathrm{F}_{(\gamma_c-1),(\gamma_u-1)} \Rightarrow u \text{ present.} \tag{3.51}$$

The test is only approximate as the mixture of three Gaussian populations with different means in the control set is not Gaussian (see Figures 3.2 and 3.3 above). However, the presence of AWGN in the original intercept results in a closer approximation to Gaussian. Note that the test is still valid if the two populations of means θ_{-1} and θ_1 are unequal as the average distance of any TCF value from $-1/L$ is the same $d = \{\mathrm{t}(n) - 1\}/L$ in both cases.

As the partial TCF of a Gold code at peaks for one of the constituent m-sequences reduces to the partial TCF of the other m-sequence at non-peak locations (3.39), its extra variance due to AWGN from $\mathrm{N}(0, \sigma^2)$ is $(\sigma^6 + 3\sigma^4 + 3\sigma^2)/N$ [1]. Although not yet derived, the corresponding extra variance at non-peak locations is at least as great. Added noise clearly reduces the power of the F-test in (3.51).

As an example, consider the case: $L = 31$, $N = 20$, SNR = 0 ($\sigma^2 = 1$), $\gamma = 0.5$. The separate variances are

$$\sigma_0^2 = \frac{1}{N}\left(1 - \frac{N-1}{L}\right) = 0.019 \qquad \text{[due to partial sample]},$$

$$\gamma\left\{\frac{\mathrm{t}(n)-1}{L}\right\}^2 = 0.034 \qquad \text{[extra due to non - peak]}, \tag{3.52}$$

$$\frac{\sigma^6 + 3\sigma^4 + 3\sigma^2}{N} = 0.35 \qquad \text{[AWGN]}.$$

A small set of simulations resulted in a constituent m-sequence detection rate of only 35% at 0 dB, but 85% at 3 dB, though these rates are pessimistic. 20 TCF values were used in both the peak set and control set, so the F-value used was $\mathrm{F}_{19,19} = 1.8$. A reduction in the 95% confidence level of the test would improve the detection rate at the expense of more false alarms. A larger N or multiple intercepts would improve the detection rate.

The use of kurtosis is a powerful aid in detecting mixtures of Gaussian distributions. Gold code detection depends on distinguishing the distributions in Figures 3.2 and 3.3: the locally most powerful test that a sample is drawn from a mixture of Gaussian distributions against the hypothesis that it is drawn from a single Gaussian population is based on kurtosis, μ_4/μ_2^2, where μ_j is the j-th moment of the sample about the mean. Such a test is being investigated. Also, as the non-peak variance (3.48, 3.49) depends on n and L, methods to narrow the search for m-sequences by using variance information to limit the range for L are also under investigation: prior knowledge of SNR is valuable.

Once a constituent m-sequence has been detected, a search is carried out for one of the few other m-sequences that may complete a preferred pair. The phases of the m-sequences do not affect the above detection process. However, the Gold code is finally identified by maximizing the correlation between the intercept and relatively phase-shifted versions of the detected preferred pair of m-sequences.

4. Higher-order (>3) Correlation Functions of Gold Codes

The set of L^2 cyclic rotations of all L of the Gold codes generated by a preferred pair of m-sequences of length L, $T^j w^k (0 \le j \le L-1,\ 0 \le k \le L-1)$, is closed with respect to $\times\{1, -1\}$:

$$
\begin{aligned}
w_i^k w_{i+n}^\ell &= u_i v_{i+k} u_{i+n} v_{i+n+\ell} \\
&= u_i u_{i+n} v_{i+k} v_{i+n+k} \\
&= u_{i+\alpha} v_{i+\beta} \\
&= w_{i+j}^{\beta-\alpha}
\end{aligned}
\tag{4.1}
$$

i.e. $\quad w_i^k w_{i+n}^\ell = w_{i+j}^r$

It follows that, in general, higher-order correlation functions are 3-valued. It has already been shown that the triple-correlation function is 3-valued and has no peaks of value 1 except for rare special cases (3.4.1). For order 4:

$$
\begin{aligned}
C^k(p,q,r) &= \frac{1}{L}\sum_i w_i^k w_{i+p}^k w_{i+q}^k w_{i+r}^k \\
&= \frac{1}{L}\sum_i w_{i+n_p}^\alpha w_{i+n_{qr}}^\beta \\
&= \frac{1}{L}\sum_i w_{i+j}^\ell
\end{aligned}
\tag{4.2}
$$

For order 5:

$$
\begin{aligned}
C^k(p,q,r,s) &= \frac{1}{L}\sum_i w_i^k w_{i+p}^k w_{i+q}^k w_{i+r}^k w_{i+s}^k \\
&= \frac{1}{L}\sum_i w_{i+n_p}^\alpha w_{i+n_{qr}}^\beta w_{i+s}^k \\
&= \frac{1}{L}\sum_i w_{i+j}^\ell w_{i+s}^k \\
&= \frac{1}{L}\sum_i w_{i+k}^m
\end{aligned}
\tag{4.3}
$$

While the above correlations are, in general, 3-valued, a 4th value of 1 would result if, for example, $P = k$ and $j = s$ in (4.3). However, such values are impossible, except in cases when $L = 3N$, until 9th-order correlation functions are reached.

4.1 9^{th}-order Correlation Functions (NCFs)

For the Gold code $w^k\left(w_i^k = u_i v_{i+k}\right)$, let (p,q) and (r,s) be respective TCF peak locations for m-sequences u and v. Then the following 9^{th}-order correlation function (NCF) values are peaks:

$$C^k(p,q,r,p+r,q+r,s,p+s,q+s) = \frac{1}{L}\sum_{i=0}^{L-1} w_i^k w_{i+p}^k w_{i+q}^k w_{i+r}^k w_{i+p+r}^k w_{i+q+r}^k w_{i+s}^k w_{i+p+s}^k w_{i+q+s}^k \tag{4.4}$$

But (p,q) is a TCF peak location for u and not for v, so

$$\begin{aligned} w_i^k w_{i+p}^k w_{i+q}^k &= u_i v_{i+k} u_{i+p} v_{i+k+p} u_{i+q} v_{i+k+q} \\ &= \left(u_i u_{i+p} u_{i+q}\right) v_{i+k} v_{i+k+p} v_{i+k+q} \\ &= v_{i+k} v_{i+k+p} v_{i+k+q} \\ &= v_{i+j}, \qquad j \neq 0 \end{aligned} \tag{4.5}$$

Also

$$\begin{aligned} w_{i+r}^k w_{i+p+r}^k w_{i+q+r}^k &= u_{i+r} v_{i+k+r} u_{i+p+r} v_{i+k+p+r} u_{i+q+r} v_{i+k+q+r} \\ &= \left(u_{i+r} u_{i+r+p} u_{i+r+q}\right) v_{i+k+r} v_{i+k+p+r} v_{i+k+q+r} \\ &= v_{i+k+r} v_{i+k+p+r} v_{i+k+q+r} \\ &= v_{i+j+r} \qquad \text{from (4.5)} \end{aligned} \tag{4.6}$$

and

$$\begin{aligned} w_{i+s}^k w_{i+p+s}^k w_{i+q+s}^k &= u_{i+s} v_{i+k+s} u_{i+p+s} v_{i+k+p+s} u_{i+q+s} v_{i+k+q+s} \\ &= \left(u_{i+s} u_{i+s+p} u_{i+s+q}\right) v_{i+k+s} v_{i+k+p+s} v_{i+k+q+s} \\ &= v_{i+k+s} v_{i+k+p+s} v_{i+k+q+s} \\ &= v_{i+j+s} \qquad \text{from (4.5)} \end{aligned} \tag{4.7}$$

As (r,s) is a TCF peak for v, (4.5), (4.6) and (4.7) result in the following expression for the NCF:

$$\begin{aligned} C^k(p,q,r,p+r,q+r,s,p+s,q+s) &= \frac{1}{L}\sum_{i=0}^{L-1} v_{i+j} v_{i+j+r} v_{i+j+s} \\ &= 1 \end{aligned} \tag{4.8}$$

These NCF peaks occur for any (p,q) TCF peak of u and any (r,s) TCF peak of v and for any permutation of p, q, r, $p+r$, $q+r$, s, $p+s$, $q+s$ in $C^k(\,)$. All permutations yield identical values and do not increase the power of tests for Gold codes constructed from u and v. However, each combination of the $(L-1)/2$ independent u peaks and the $(L-1)/2$ independent v peaks yield independent Gold code peaks, a total of $(L-1)^2/4$. Thus if (p_i,q_i), $i = 1,2,.., (L-1)/2$, and (r_j,s_j), $j = 1,2,.., (L-1)/2$ are the respective locations of u and v TCF-peaks, each of the $(L-1)^2/4$ NCF locations:

$$(p_i, q_i, r_j, p_i + r_j, q_i + r_j, s_j, p_i + s_j, q_i + s_j), \quad 1 \le i, j \le (L-1)/2 \tag{4.9}$$

is a peak. Thus the NCF is 4-valued. The peak-values of 1 are extremely rare: the vast majority of the L^9 NCF locations produce one of the three values θ_{-1}, θ_0 or θ_1 (3.1), as in (4.1), (4.2) and (4.3).

Tests for Gold codes are based on the mean of the $(L-1)^2/4$ values at the NCF locations above, which should be peak values of 1 if a Gold code is present. If no such code from the (u,v) family is present, the expected value of the mean is $-1/L$. For each possible sequence length L, a search may be made for each preferred pair (u,v), i.e. each set $G(u,v)$. Each Gold code in $G(u,v)$ has the same NCF peaks, determined by (p_i,q_i) and (r_i,s_i), so the search space is small.

4.2 Statistics of the NCF of Gold codes in noise

Gold code w of length L in additive Gaussian noise n [~N(0, σ^2)] has NCF mean:

$$\begin{aligned}
\overline{C}_9(j_1, j_2, \ldots, j_8) &= \mathrm{E}\left[\frac{1}{L}\sum_{i=1}^{L}(w_i + n_i)(w_{i+j_1} + n_{i+j_1})\ldots(w_{i+j_8} + n_{i+j_8})\right] \\
&= \mathrm{E}\Bigg[\frac{1}{L}\sum_{i=1}^{L}\Big\{ w_i w_{i+j_1}\cdots w_{i+j_8} \\
&\qquad + n_i w_{i+j_1}\cdots w_{i+j_8} \\
&\qquad + n_{i+j_1} w_i w_{i+j_2}\cdots w_{i+j_8} \\
&\qquad \vdots \\
&\qquad + n_{i+j_8} w_i w_{i+j_1}\cdots w_{i+j_7} \\
&\qquad + n_i n_{i+j_1} w_{i+j_2}\cdots w_{i+j_8} \\
&\qquad \vdots \\
&\qquad + n_i n_{i+j_1} n_{i+j_2} w_{i+j_3}\cdots w_{i+j_8} \\
&\qquad \vdots \\
&\qquad + n_i n_{i+j_1}\ldots n_{i+j_8}\Big\}\Bigg] \\
&= \mathrm{E}\left[\frac{1}{L}\sum_{i=1}^{L} w_i w_{i+j_1}\cdots w_{i+j_8}\right]
\end{aligned} \tag{4.10}$$

as all other terms involve at least one n_k (and no n_k^m for $m > 1$) for $k \varepsilon S = \{i, i+j_1, \ldots, i+j_8\}$ and

$$\begin{aligned}
&\mathrm{E}\left[n_k f\left(n_{l_1} n_{l_2} \ldots \mid l_m \varepsilon\, S \text{ and } l_m \neq k\right)\right] \\
&= \mathrm{E}[n_k]\,\mathrm{E}\left[f\left(n_{l_1} n_{l_2} \ldots\right)\right] = 0
\end{aligned} \tag{4.11}$$

Thus the following (unbiased) NCF-means result for peak and non-peak locations:

$$\overline{C}_9(j_1, j_2, \ldots, j_8) = \begin{cases} 1 & \text{if } (j_1, j_2, \ldots, j_8) \text{ is a NCF peak} \\ -1/L & \text{if } (j_1, j_2, \ldots, j_8) \text{ is not a NCF peak} \end{cases} \tag{4.12}$$

If $(j_1, j_2, \ldots, j_8)$ is a NCF-peak location:

$$\begin{aligned}
w_i w_{i+j_1} \ldots w_{i+j_8} = 1 \Rightarrow\ & n_i w_{i+j_1} \ldots w_{i+j_8} = n_i w_i \\
& n_{i+j_1} w_i w_{i+j_2} \ldots w_{i+j_8} = n_{i+j_1} w_{i+j_1} \\
& \vdots \\
& n_{i+j_8} w_i w_{i+j_1} \ldots w_{i+j_7} = n_{i+j_8} w_{i+j_8} \\
& n_i n_{i+j_1} w_{i+j_2} \ldots w_{i+j_8} = n_i n_{i+j_1} w_i w_{i+j_1} \\
& \vdots
\end{aligned} \tag{4.13}$$

The variance of the NCF of $w + n$ at these peak locations is

$$\begin{aligned}
\operatorname{var}[C_9(j_1, j_2, \ldots, j_8)] &= \mathrm{E}\left[\left\{\frac{1}{L}\sum_{i=1}^{L}(w_i + n_i)(w_{i+j_1} + n_{i+j_1})\ldots(w_{i+j_8} + n_{i+j_8}) - \overline{C}_9\right\}^2\right] \\
&= \mathrm{E}\left[\frac{1}{L^2}\left\{\sum_{i=1}^{L}(w_i + n_i)(w_{i+j_1} + n_{i+j_1})\ldots(w_{i+j_8} + n_{i+j_8})\right\}^2\right] - \overline{C}_9^{\,2} \\
&= \frac{1}{L^2}\mathrm{E}\Big[\Big\{w_1 w_{i+j_1} \ldots w_{i+j_8} + \ldots + w_L w_{L+j_1} \ldots w_{L+j_8} \\
&\qquad + n_1 w_1 + \ldots + n_L w_L \\
&\qquad + n_{1+j_1} w_{1+j_1} + \ldots + n_{L+j_1} w_{L+j_1} \\
&\qquad \vdots \\
&\qquad + n_{1+j_8} w_{1+j_8} + \ldots + n_{L+j_8} w_{L+j_8} \\
&\qquad + n_1 n_{1+j_1} w_1 w_{1+j_1} + \ldots + n_L n_{L+j_1} w_L w_{L+j_1} \\
&\qquad + n_1 n_{1+j_2} w_1 w_{1+j_2} + \ldots + n_L n_{L+j_2} w_L w_{L+j_2} \\
&\qquad \vdots \\
&\qquad + n_1 n_{1+j_1} \ldots n_{1+j_8} + n_2 n_{2+j_1} \ldots n_{2+j_8} + \ldots + n_L n_{L+j_1} \ldots n_{L+j_8}\Big\}^2\Big] - 1
\end{aligned} \tag{4.14}$$

Squaring E[] in (4.14) and omitting all cross-product terms involving single n_k expressions for any k, which have zero expectation, gives the following variance:

$$
\begin{aligned}
\operatorname{var}[C_9(j_1, j_2, \ldots, j_8)] = \frac{1}{L^2}\mathrm{E}\Big[L^2 & \qquad [L^2] \\
&+ w_1 w_2 w_{1+j_1} w_{2+j_1} \ldots w_{1+j_8} w_{2+j_8} + w_1 w_3 \ldots + \ldots \qquad [0] \\
&+ \sum_{i=1}^{L} n_i^2 w_i^2 + \sum_{i=1}^{L} n_{i+j_1}^2 w_{i+j_1}^2 + \ldots + \sum_{i=1}^{L} n_{i+j_8}^2 w_{i+j_8}^2 \qquad [9L\sigma^2] \\
&\left.\begin{aligned}
&+ \sum_{i=1}^{L} n_i^2 n_{i+j_1}^2 w_i^2 w_{i+j_1}^2 \qquad [L\sigma^4] \\
&+ \sum_{i=1}^{L} n_i^2 n_{i+j_2}^2 w_i^2 w_{i+j_2}^2 \qquad [L\sigma^4] \\
&+ \\
&\vdots \\
&+ \sum_{i=1}^{L} n_{i+j_7}^2 n_{i+j_8}^2 w_{i+j_7}^2 w_{i+j_8}^2 \qquad [L\sigma^4]
\end{aligned}\right\} \times \frac{9!}{7!2!} \\
&\left.\begin{aligned}
&+ \sum_{i=1}^{L} n_i^2 n_{i+j_1}^2 n_{i+j_2}^2 w_i^2 w_{i+j_1}^2 w_{i+j_2}^2 \qquad [L\sigma^6] \\
&+ \\
&\vdots \\
&+ \sum_{i=1}^{L} n_{i+j_6}^2 n_{i+j_7}^2 n_{i+j_8}^2 w_{i+j_6}^2 w_{i+j_7}^2 w_{i+j_8}^2 \qquad [L\sigma^6]
\end{aligned}\right\} \times \frac{9!}{6!3!} \\
&+ \\
&\vdots \\
&+ \sum_{i=1}^{L} n_i^2 n_{i+j_1}^2 \ldots n_{i+j_8}^2 \Big] \qquad [L\sigma^{18}] \\
&-1 \\
= \frac{1}{L^2}&\left\{L^2 + 9L\sigma^2 + \frac{9!}{2!7!}L\sigma^4 + \frac{9!}{3!6!}L\sigma^6 + \ldots + L\sigma^{18}\right\} - 1 \\
= \frac{1}{L}&\sum_{k=1}^{9} \frac{9!}{k!(9-k)!}\sigma^{2k}
\end{aligned}
\tag{4.15}
$$

The variance of the NCF at peak (and also non-peak) locations is thus, in full:

$$
\begin{aligned}
&\operatorname{var}[C_9(j_1, j_2, \ldots, j_8)] = \\
&\quad \frac{1}{L}\left\{9\sigma^2 + 36\sigma^4 + 84\sigma^6 + 126\sigma^8 + 126\sigma^{10} + 84\sigma^{12} + 36\sigma^{14} + 9\sigma^{16} + \sigma^{18}\right\}
\end{aligned}
\tag{4.16}
$$

5. Detection Based on Triple Products

5.1 Detection of m-sequences

An m-sequence u may be detected by the property that, if $(-p', -q')$ is a TCF-peak location, the triple product $u_i u_{i-p'} u_{i-q'}$ is always equal to one (see Figure 5.1):

$$\forall i.\ u_i = u_{i-p'} u_{i-q'} \Rightarrow u_i u_{i-p'} u_{i-q'} = 1 \tag{5.1}$$

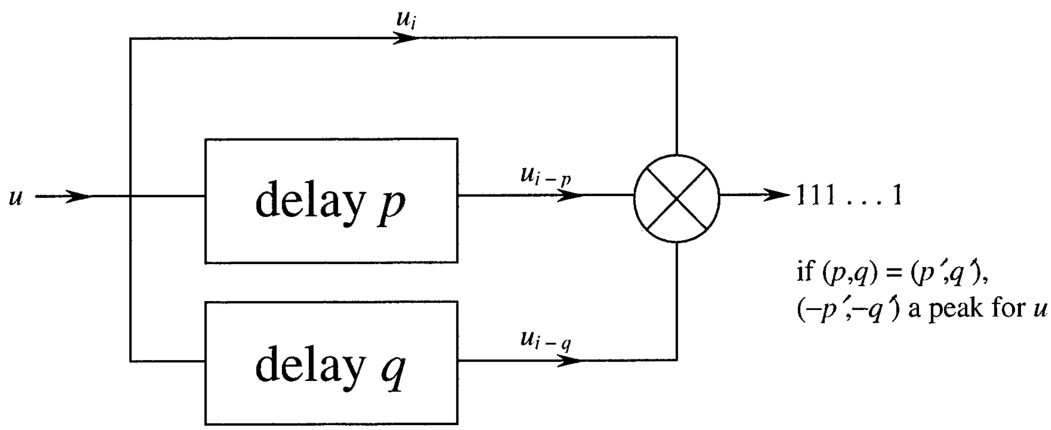


Figure 5.1 Triple-product based m-sequence detector (suggested by R Gold)

Each m-sequence (/ generator polynomial) of length L may be represented by a single peak location, and each location examined by testing the hypothesis that the outputs in Figure 5.1 are one.

If the m-sequence is data modulated by b_i, which changes every Mth value ($M = mL$), i.e.

$$b_i = \pm 1,\ \ jM + 1 \leq i \leq (j+1)M,\ \ 0 \leq j \leq N-1, \tag{5.2}$$

then the output of the detector (Figure 5.1) for (p,q) corresponding to a TCF u-peak, and assuming no noise, is as shown in Figure 5.2.

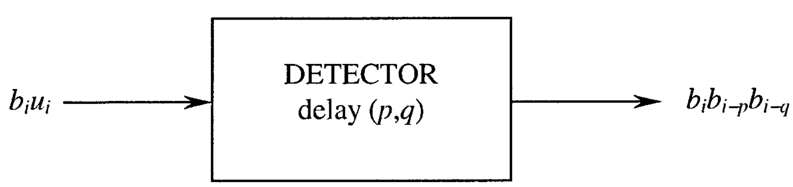


Figure 5.2 Output of m-sequence detector after data modulation

The output $b_i b_{i-p} b_{i-q}$ closely resembles the data signal b_i as $(+1)^3 = +1$ and $(-1)^3 = -1$. Assuming $q>p$, errors may occur in relatively few b_i within p and q values after a data boundary where the previous data value is −1, as illustrated in Figure 5.3. The representative peak should be chosen to ensure $q-p$ is as small as possible: although $q \neq p$ (no such peaks exist), it is possible that $q-p = 1$. E.g., for $L = 31$, [45] has a peak (17,18) and [75] has a peak (19,20), for both of which $q - p = 1$. Consequently, the notch shown in the rectangular pulse in Figure 5.3 would be only one chip wide. For longer m-sequences, the effects of such errors would become negligible.

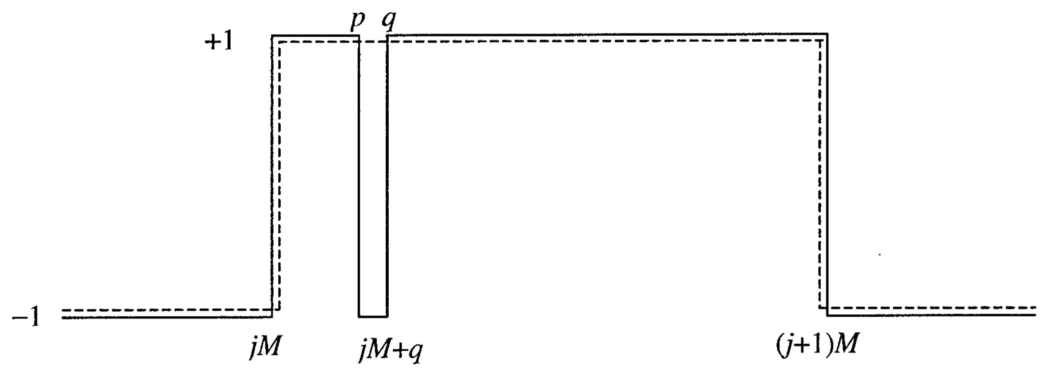


Figure 5.3 Output approximation to data b_i

The detector thus also acts as a decoder which does not require phase acquisition. Such a self-reference decoder would be dependent only on the chosen shift pair for the particular code and not on the chip rate. It would also possess LPD/LPI by chip-rate hopping – pseudo-randomly varying the chip rate. This idea has not been developed here but seems worthy of future investigation.

5.1.1 **Mean and Variance of m-Sequence Detector Output**

Assuming addive Gaussian noise $n_i \sim N(0, \sigma^2)$, the mean detector output is

$$
\begin{aligned}
&\mathrm{E}[\ (b_i u_i + n_i)(b_{i-p} u_{i-p} + n_{i-p})(b_{i-q} u_{i-q} + n_{i-q})\ ] \\
&\quad = \mathrm{E}[\ b_i b_{i-p} b_{i-q} u_i u_{i-p} u_{i-q} \\
&\qquad + n_i b_{i-p} u_{i-p} b_{i-q} u_{i-q} + n_{i-p} b_i u_i b_{i-q} u_{i-q} + n_{i-q} b_i u_i b_{i-p} u_{i-p} \\
&\qquad + n_i n_{i-p} b_{i-q} u_{i-q} + n_i n_{i-q} b_{i-p} u_{i-p} + n_{i-p} n_{i-q} b_i u_i \\
&\qquad + n_i n_{i-p} n_{i-q}\ ] \\
&\quad = b_i b_{i-p} b_{i-q} \qquad \{(p,q) \text{ corresponds to a peak}\}
\end{aligned}
\tag{5.3}
$$

Note that, in the case of no data modulation $(\forall i \bullet b_i = 1)$, the mean is 1.

As cross products vanish, the variance of the detector output is

$$
\mathrm{E}[n_i^2 + n_{i-p}^2 + n_{i-q}^2 + n_i^2 n_{i-p}^2 + n_i^2 n_{i-q}^2 + n_{i-p}^2 n_{i-q}^2 + n_i^2 n_{i-p}^2 n_{i-q}^2] = 3\sigma^2 + 3\sigma^4 + \sigma^6
\tag{5.4}
$$

This is the variance of each output value, irrespective of whether the input m-sequence is modulated or (p,q) corresponds to a TCF peak. It is greater than the TCF variance in [1, pA.8] by a factor of L, as expected, as the TCF is the mean of L triple products.

5.2 Detection of Gold Codes

A Gold code $w = uv$, where u and v are m-sequences with respective TCF peaks $(-p',-q')$ and $(-r',-s')$, may be detected by testing if the outputs of the 2-stage detector in Figure 5.2 are equal to one.

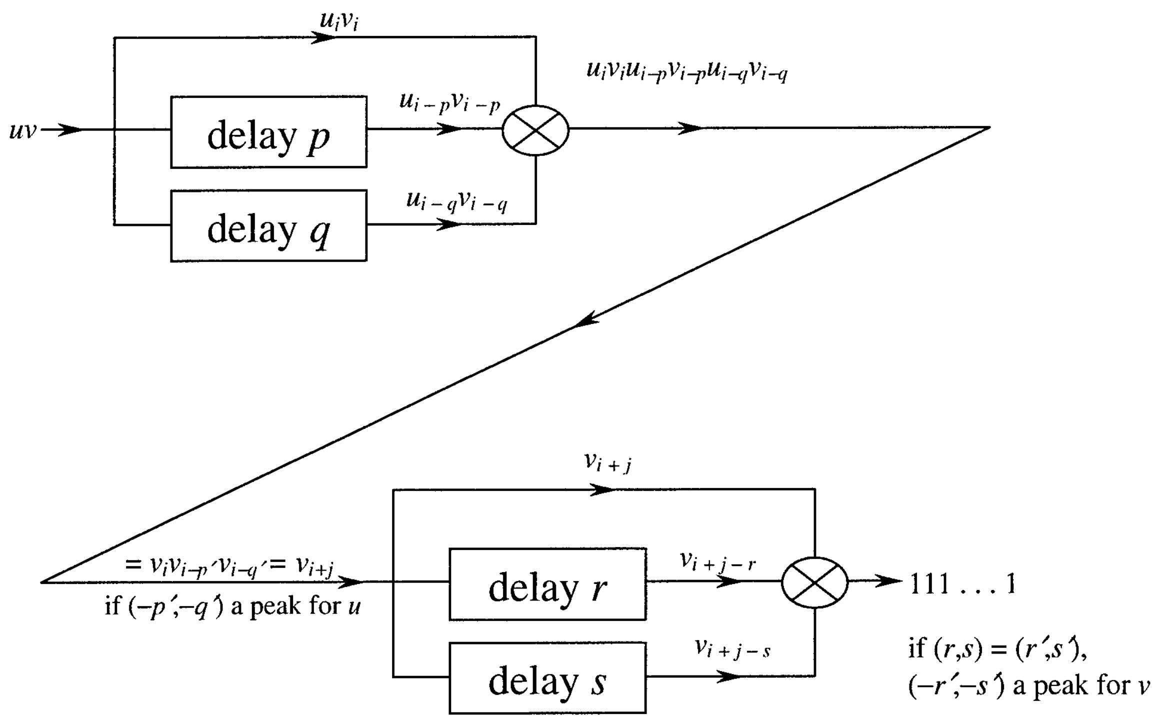


Figure 5.4 2-stage triple-product based Gold code detector (suggested by R Gold)

u and v may be found by a 2-dimensional search for (p',q') and (r',s'):

$$(-p',-q') \text{ a peak for } u \;\Rightarrow\; \text{Stage-1 output } u_i v_i u_{i-p'} v_{i-p'} u_{i-q'} v_{i-q'} = v_i v_{i-p'} v_{i-q'} \tag{5.5}$$

$$\text{Group closure} \;\Rightarrow\; v_i\, v_{i-p'} v_{i-q'} = v_i v_{i+k} = v_{i+j}, \text{ as } (-p',-q') \text{ not a peak for } v. \tag{5.6}$$

$$\begin{aligned}(-r',-s') \text{ a peak for } v &\Rightarrow \text{also a peak for } j\text{-shifted } v_{i+j}, \text{ by TCF phase-invariance}\\ &\Rightarrow \forall i \bullet \text{Stage-2 output } v_{i+j} v_{i+j-r'} v_{i+j-s'} = 1.\end{aligned} \tag{5.7}$$

If the Gold code is data modulated with b_i, as in (5.2), then the noiseless output of the 2-stage detector (Figure 5.4) for (p,q) and (r,s) values corresponding to peaks is as shown in Figure 5.5.

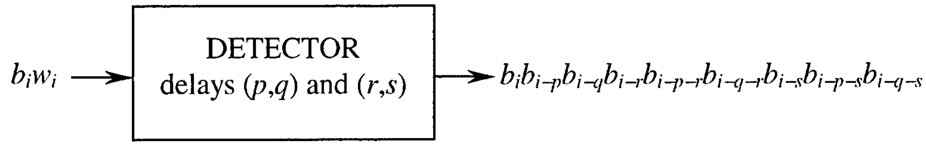


Figure 5.5 Output of Gold code detector after data modulation

The output $b_i b_{i-p} b_{i-q} b_{i-r} b_{i-p-r} b_{i-q-r} b_{i-s} b_{i-p-s} b_{i-q-s}$ closely resembles the data signal b_i as $(+1)^9 = +1$ and $(-1)^9 = -1$. Error patterns are more complicated than those for the m-sequence detector illustrated in Figure 5.3. An example is given in Figure 5.6 for preferred pair [45] and [75] with $(p,q) = (17,18)$ and $(r,s) = (19,20)$. Again, the effect of these errors would be greatly reduced for longer sequences. A decoder similar to that suggested in 5.1 would suffer from greatly increased noise.

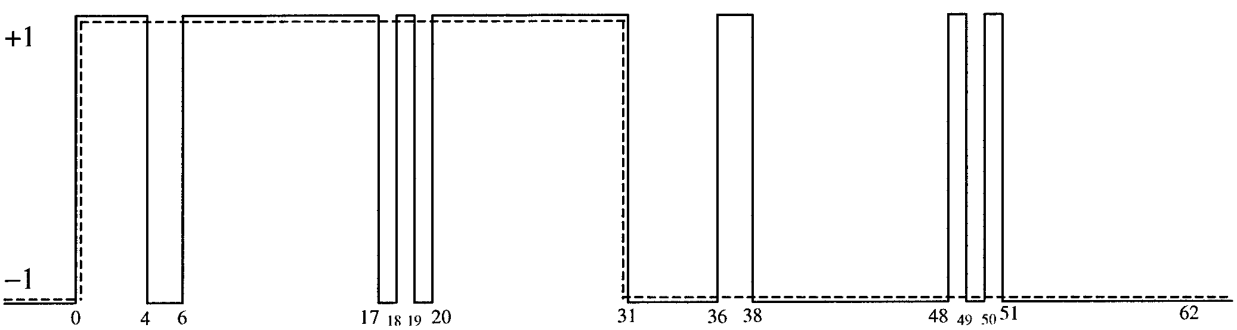


Figure 5.6 Output of 2-stage detector as approximation to data b_i

5.2.1 Mean and Variance of Gold Code 2-stage Detector Output

Assuming noise $n_i \sim \mathrm{N}(0, \sigma^2)$, the mean of the detector output d_i is

$$\begin{aligned}\bar{d}_i &= \mathrm{E}\left[(b_i w_i + n_i)(b_{i-p} w_{i-p} + n_{i-p})\ldots(b_{i-q-s} w_{i-q-s} + n_{i-q-s})\right] \\ &= b_i b_{i-p} b_{i-q} b_{i-r} b_{i-p-r} b_{i-q-r} b_{i-s} b_{i-p-s} b_{i-q-s}\end{aligned} \tag{5.8}$$

For no data modulation, this value is always 1.

The variance is $L\times$ greater than that of the NCF variance (4.16):

$$\mathrm{var}[d_i] = 9\sigma^2 + 36\sigma^4 + 84\sigma^6 + 126\sigma^8 + 126\sigma^{10} + 84\sigma^{12} + 36\sigma^{14} + 9\sigma^{16} + \sigma^{18} \tag{5.9}$$

5.2.2 Statistical Tests to Detect Gold Codes

Clearly, the high variance in (5.9) suggest the averaging of many output values to test possible (p,q) and (r,s) pairs. If data modulation is present, such a test may be based on output spectra. Assuming a chip duration τ, data modulation interval $M\tau$ and b_i random, the expected Fourier transform (FT) is $M\tau\mathrm{sinc}\,(fM\tau)$, ignoring the effect of the signal's error notches. The power spectrum envelope of the 2-stage-detector output for correct (p,q) and (r,s) pairs is approximately as shown in Figure 5.6.

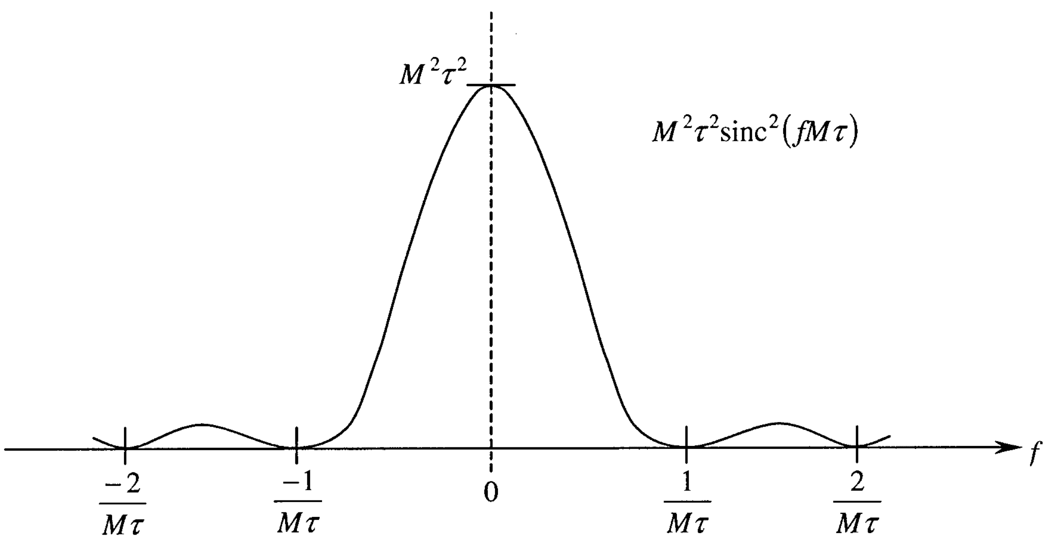


Figure 5.7 Output power spectrum for correct (p,q), (r,s)

When (p,q) and (r,s) are not peaks, the output of the 2-stage detector for unmodulated input is

$$w_i^k w_{i-p}^k w_{i-q}^k w_{i-r}^k w_{i-p-r}^k w_{i-q-r}^k w_{i-s}^k w_{i-p-s}^k w_{i-q-s}^k = w_{i-j}^{\ell} \tag{5.10}$$

for some ℓ and j ($0 \le \ell, j \le L-1$), from the closure law (4.1), i.e. a shifted Gold code from the set $G(u,v)$ (3.7). The output for a b-modulated Gold code is thus

$$\mu_i = w_{i-j}^{\ell} b_i b_{i-p} b_{i-q} b_{i-r} b_{i-p-r} b_{i-q-r} b_{i-s} b_{i-p-s} b_{i-q-s}, \tag{5.11}$$

which may be approximated by

$$\mu_i' = w_{i-j}^{\ell} b_i \,. \tag{5.12}$$

Now the FT of μ_i' is given by the convolution of FTs:

$$\mathrm{FT}\left(w_{i-j}^{\ell} b_i\right) = W_{i-j}^{\ell} * M\tau \,\mathrm{sinc}(fM\tau), \tag{5.13}$$

where the Gold code FT, W_{i-j}^{ℓ}, has zeros at k/τ, $k = \pm 1, \pm 2, \ldots$. There is no general expression for W_{i-j}^{ℓ}: Gold codes of the same length have different FTs and power spectra. However, the FT in (5.13) has zeros at k/τ rather than the $k/M\tau$ for peak (p,q), (r,s) values. This feature, in addition to the different expected values at zero frequency, is the basis for selecting (p,q) and (r,s) corresponding to a particular preferred (u,v) pair/$G(u,v)$ set. Smoothing the detector's output, to reduce the effect of error notches, would reduce noise in the estimated power spectrum.

For correct shifts, the output may be approximated by a random signal (± 1), changing at $M\tau$ intervals, with power spectrum $M^2\tau^2 \,\mathrm{sinc}^2(fM\tau)$. Incorrect shifts produce an approximately pseudo-random (± 1) output, changing at τ intervals, with approximate power spectrum $\tau^2 \,\mathrm{sinc}^2(f\tau)$. This is illustrated in the graphs of power spectra following listings of programs Goldetmods and Goldetmodsn in the appendix. Power at 0 frequency is consistently higher for correct shifts corresponding to peaks, even at –6 dB. This feature offers a good test for the presence of a modulated Gold code. Among the tests under investigation is one based on the statistic

$$\gamma = \frac{\sum_i x_i / f_i}{\sum_i x_i} \tag{5.14}$$

where x_i are power-spectrum values and f_i are corresponding frequencies. γ has high expected values for correct shifts.

6. Results

6.1 Tests based on Gold code TCFs

Tests were based on the differences in the statistical moments of TCF values on peak sets (sets of TCF peak locations for either u or v) with those on non-peak sets, for $L = 31$. Kurtosis produced very poor discrimination. The mean alone was better: peak sets had TCF means close to $-1/31 = -0.032$, whereas non-peak sets had means close to -0.075 due to larger numbers of $-9/31$ than $7/31$. Variance was always lower for peak sets, as was skewness, again due to the imbalance of $-9/31$ and $7/31$ values in non-peak sets. A linear combination of the first three moments produced the detection/false-alarm rates in Table 6.1, using single 31-length samples. The detection rates may be improved by averaging over several samples.

Noise level	Detection rate	False-alarm rate
6 dB	98	1
3 dB	67	25
0 dB	56	34

Table 6.1 TCF moment-based detection and false-alarm % for Gold codes

6.2 Tests based on Gold code NCFs

Extensive simulations were carried out to determine detection and false-alarm rates for 31-length Gold codes with varying noise levels. The preferred pair of m-sequences used were $u[45]$ and $v[75]$, as defined in (3.10) and (3.12). The % detection and false alarm rates in Table 6.2 are based on single 31-length samples of the Gold code in noise. All combinations of the two sets of m-sequence TCF-peak locations are used in the tests, as described in section 4.1, producing a mean of 225 NCF values. This mean μ is compared with a threshold of 0.9, chosen because of upward bias of NCF values due to higher-order statistical properties of the noise generator used.

The hypotheses tested were:

H_0: Gold sequence from $G(u,v)$ not present (if $\mu \leq 0.9$)

H_1: Gold sequence from $G(u,v)$ present (if $\mu > 0.9$)

The % detection/false-alarm rates are derived from 1000 simulations at each noise level. For false-alarms rates, non-peak locations were used, the recorded rate being the % of mean values exceeding 0.9. Note that, using (4.16), the approximate variance of NCF values is 0.27, consistent with observed values in the simulations.

Noise level	Detection rate	False-alarm rate
3 dB	100	0
0 dB	99.3	0.0
−3 dB	77.9	26.2
−6 dB	58.3	46.6

Table 6.2 Gold code detection and false alarm % based on NCFs of 31-samples

Further simulations were carried out basing detection and false alarms on averages of NCF values over 10 independent 31-length samples. The improved % rates are shown in Table 6.3.

Noise level	Detection rate	False-alarm rate
3 dB	100	0
0 dB	100	0
−3 dB	100	7
−6 dB	74	43

Table 6.3 Gold code detection and false-alarm % based on 10-averages of 31-sample NCFs

For 127-length Gold codes, the increased length and extra NCF locations, reducing variance by a factor of 72, makes detection easier. From (4.16), at −3 dB variance is increased by a factor of 38.5, so 100% detection and 0% false-alarms are expected. At −6 dB, variance is increased (4.16) by a factor of 3822, which means 53 127-samples must be averaged to achieve the 0 dB performance for 31-length sequences, i.e. 99% detection, 0% false-alarms.

6.3 Tests based on 2-stage triple products of Gold codes

A repeating Gold code was generated from the same preferred pair (u,v), as in 6.2 with independent additive noise. A similar upward bias to that described in 6.2 was noted in the mean of the 2-stage detector output of 5.2. The same threshold of 0.9 was therefore selected. Assuming μ_N is the mean output over N successive values, the following hypotheses were tested:

H_0 : (p,q) and (r,s) correct (if $\mu_N > 0.9$)

H_1 : (p,q) and (r,s) incorrect (if $\mu_N \leq 0.9$)

The detection % shown in Table 6.4 is calculated from the number of values >0.9 in 200 simulations when correct shift pairs are used; the false-alarm % is from the number of values >0.9 when the shift pairs are incorrect for the input Gold code.

N = number of outputs averaged	Detection rate	False-alarm rate
500	77.5	35.5
1000	85.5	30.0
2000	93.0	22.5

Table 6.4 Detection and false alarm % from output of

2-state detector for 0 dB noise

Clearly, from Table 6.4, averages of many output values are necessary to achieve moderate detection rates, even at 0 dB. As output variance is greater than that of the NCF estimator by a factor of $L(L-1)^2/4$, for $L = 31$ approximately 7000 output values must be averaged to match the NCF detection rate of 99% at 0 dB given in Table 6.2. As infeasibly large averages would be required at higher noise levels, no further simulations were made.

7. Conclusions

The TCF method for Gold code detection in 3.4 has the advantage of low complexity and is two-stage. The TCF locations for possible m-sequences of appropriate length are examined separately. When it is decided that an m-sequence u is one of the preferred pair from which the Gold code is constructed, the search for a preferred complimentary v is narrowed down: it is not necessary to simultaneously find u and v. Phases of u and v are also irrelevant, reducing the dimensionality of the search. However, the method is very sensitive to noise and, for good detection rates, many L-samples are necessary to allow TCF averaging. The noise problem reduces for longer Gold codes as each TCF value results from larger triple averages and more TCF values are available from each L-sample.

The NCF method is the most successful overall. The lowest-order correlation function with Gold code peaks is nine. Although very noisy, there are $(L-1)^2/4$ NCF peak locations to average over, and each NCF value is an average of L ninth-order products. Thus the computational load is not high as the NCF is only evaluated at a small number of locations for each possible preferred-pair. Even for L=31 at 0 dB, 99.3% detection of Gold codes is possible, with 0.0% false-alarms, from a single 31-sample, and increasing L rapidly improves detection. However, detection rates fall sharply for higher noise levels. All preferred pairs (u,v) for each L must be tried, but phases of u,v and the Gold code are irrelevant.

The third method, a 2-stage triple-correlator detector, has the virtue of simplicity and is easily implementable in hardware. Three relatively shifted versions of the input are multiplied to produce an output of triple products, three relatively shifted versions of which producing the output of the 2nd stage. As single triples are used, the output is extremely noisy and very large numbers must be averaged to produce moderate detection at even low noise levels. For data-modulated Gold codes, spectral tests for codes are possible: spectra of outputs for correct shift pairs are easily distinguishable from spectra for incorrect shift pairs from their increased low-frequency power. In principle, the 2-stage detector may also be used as a phase-independent decoder, but the noisy output would restrict its use to long codes.

The NCF and correlator detectors in sections 4 and 5 have the advantage that the presence of further codes from the set $G(u,v)$ strengthen statistical tests, whereas the statistical properties of the TCF discriminants of section 3 are undermined.

8. **References.**

[1] E R Adams and P C J Hill, Detection of covert DS/SS signals using higher-order statistical processing, Wright Lab Tech Report WL-TR-97-1119 for contract F61708 95 C008, 1997.

[2] K E Batty, Detection and characterisation of direct sequence spread spectrum signals using higher order statistics, MSc Thesis, RMCS (Cranfield University), 1998.

[3] K E Batty and E R Adams, Detection and blind identification of m-sequence codes using higher-order statistics, Proc IEEE Signal Processing Workshop on Higher-order Statistics, Israel, ISBN 0-7695-0140-0, June 1999.

[4] D V Sarwate and M B Pursley, Crosscorrelation properties of pseudorandom and related sequences, Proc IEEE, Vol 68/5, May 1980.

Program listing of Matlab M-files.

gold.m

```
function y = gold(N,start)
% This function generates a Gold code pseudo-random binary sequence
% of length N.
% Inputs, length of sequence start and mask
% Mask is a two row vector representing the tap points of both
% generators used in the Gold code
% start is a vector of length N and is the starting state of one of the generators
% Output is a vector, y of length N containing the Gold code

switch N
case 4
   mask = [0 0 1 1;1 0 0 1];
case 5
   mask = [0 0 0 1 1;1 0 0 0 1];
otherwise
   error('This length prbs not supported')
end
a = ones(1,N);
b = start;
if(start ==0)
   error('Starting state must not be all zero')
end
for k=1:2^N-1,
   z1 = a & mask(1, :);
   z2 = b & mask(2, :);
   for i = N: -1: 2,
     z1(i-1) = xor(z1(i),z1(i-1));
     z2(i-1) = xor(z2(i),z2(i-1));
  end
   a = [z1(1),a(1:(N-1))]; % shift right operator
   b = [z2(1),b(1:(N-1))];
   y1 = xor(a(1),b(1)); % generate Gold code here
   %
   if b(1)==1
     bbb=-1;
   else
     bbb=1;
   end
   %
   if y1(1)==1, % convert to a signal voltage ('1' = -1V, '0' = +1V)
      y(k) = -1;
   else
      y(k) = 1;
   end
end
```

Goldet.m

```
% 'Goldet'
% Gold's 2-stage method to detect Gold codes
%
% specify delay pairs (p,q) and (r,s)- possible pairs (17,18) and (19,20)
%
p=17;
q=18;
r=19;
s=20;
%
% Input noise factor fn and number of 31-length Gold code cycles nc
%
fn=0;
nc=500;
%
x=[1 1 1 -1 -1 1 -1 -1 -1 -1 1 -1 -1 1 -1 1 1 1 -1 -1 -1 -1 -1 -1 1 -1 1 1 1 1 1];
%
% Generate nc cycles of 31-length Gold code in y
%
for i=1:nc
  ia=(i-1)*31;
  for j=1:31
    y(ia+j)=x(j)+randn*fn;
  end
end
% Calculate triple product for shift pair (p,q)
%
ni=nc*31-30;
for i=1:ni
  z(i)=y(i)*y(i+p)*y(i+q);
end
%
% Calculate triple product of 1st stage output for shift pair (r,s)
%
ni=ni-30;
for i=1:ni
  t(i)=z(i)*z(i+r)*z(i+s);
end
%
for i=1:124
  c(i)=t(i);
end
disp(c)
%
% Calculate mean and variance of 2nd stage output
%
s=0;
for i=1:ni
  s=s+t(i);
end
s=s/ni;
%
ss=0;
for i=1:ni
```

```
  d=t(i)-s;
  ss=ss+(d*d);
end
ss=ss/ni;
%
disp('No. of output values averaged')
disp(ni)
disp('MEAN and VARIANCE')
disp(s)
disp(ss)
%
```

NCFGoldpeaks.m

```
% NCFGoldpeaks.m
% 9th-order-CF of Gold codes using all combinations of TCF peak locations: detection %
%
fn=1
nii=1000
nav=10
%
pq(1,1)=3;
pq(1,2)=5;
pq(2,1)=6;
pq(2,2)=10;
pq(3,1)=12;
pq(3,2)=20;
pq(4,1)=24;
pq(4,2)=9;
pq(5,1)=17;
pq(5,2)=18;
pq(6,1)=7;
pq(6,2)=16;
pq(7,1)=14;
pq(7,2)=1;
pq(8,1)=28;
pq(8,2)=2;
pq(9,1)=25;
pq(9,2)=4;
pq(10,1)=19;
pq(10,2)=8;
pq(11,1)=11;
pq(11,2)=23;
pq(12,1)=22;
pq(12,2)=15;
pq(13,1)=13;
pq(13,2)=30;
pq(14,1)=26;
pq(14,2)=29;
pq(15,1)=21;
pq(15,2)=27;
%
rs(1,1)=1;
```

```
rs(1,2)=12;
rs(2,1)=2;
rs(2,2)=24;
rs(3,1)=4;
rs(3,2)=17;
rs(4,1)=8;
rs(4,2)=3;
rs(5,1)=16;
rs(5,2)=6;
rs(6,1)=5;
rs(6,2)=28;
rs(7,1)=10;
rs(7,2)=25;
rs(8,1)=20;
rs(8,2)=19;
rs(9,1)=9;
rs(9,2)=7;
rs(10,1)=18;
rs(10,2)=14;
rs(11,1)=11;
rs(11,2)=30;
rs(12,1)=22;
rs(12,2)=29;
rs(13,1)=13;
rs(13,2)=27;
rs(14,1)=26;
rs(14,2)=23;
rs(15,1)=21;
rs(15,2)=15;
%
mns=0;
vss=0;
nod=0;
for iii=1:nii

ss=0;
%
for ipq=1:15
  for irs=1:15
% Set values p q r p+r q+r s p+s q+s
i1=pq(ipq,1);
i2=pq(ipq,2);
i3=rs(irs,1);
i4=i1+i3;
i5=i2+i3;
i6=rs(irs,2);
i7=i1+i6;
i8=i2+i6;
%
x=[1 1 1 -1 -1 1 -1 -1 -1 -1 1 -1 -1 1 -1 1 1 1 -1 -1 -1 -1 -1 -1 1 -1 1 1 1 1 1];
%
%   set x(32-62) etc to same values - sequence periodic - after adding noise
%
for i=1:31
  x(i+31)=x(i)+randn*fn;
  x(i+62)=x(i+31);
```

```
  x(i+93)=x(i+31);
end
%
%  calculate 9th-O-CF c(i1,i2,i3,i4,i5,i6,i7,i8)
%
%
s=0;
for k=1:31
  y=x(k)*x(k+i1)*x(k+i2)*x(k+i3)*x(k+i4)*x(k+i5)*x(k+i6)*x(k+i7)*x(k+i8);
  s=s+y;
end
si(ipq,irs)=s/31;
ss=ss+s/31;
end
end
%
mn=ss/225;
amn(iii)=mn;
if mn > 0.9
  nod=nod+1;
else
  nod=nod+0;
end
%
mns=mns+mn;
%
%disp(si)
%
vs=0;
for ipq=1:15
  for irs=1:15
    vs=vs+(si(ipq,irs)-mn)^2;
  end
end
vs=vs/225;
vss=vss+vs;
end
%
mns=mns/10;
vss=vss/100;
%disp('MEAN  VAR')
%disp(mns)
%disp(vss)
%
nod=100*nod/nii;
disp('Detection %')
disp(nod)
%disp(amn)
disp('****************************************')
anii=nii/nav;
nod=0;
%
for ii=1:anii
  s=0;
  n=nav*(ii-1);
  for i=n+1:n+nav
```

```
    s=s+amn(i);
  end
  s=s/nav;
  bmn(ii)=s;
  if s > 0.9
    nod=nod+1;
  else
    nod=nod+0;
  end
end
%
disp('Detection % based on nav-averages')
nod=100*nod/anii;
disp(nod)
disp(bmn)
%
```

NCFGoldpeaksm.m

```
% NCFGoldpeaksm.m
% 9th-order-CF of Gold codes using all combinations of TCF peak locations: detection %
% based on NCF mean for all expected peak locations.
% 10 Gold codes used from same set G(u,v).
%
clear all
fn=0.5
nii=100
nav=10
%
N=5;
L=2^N-1;
start1=[1 1 1 1 1];%Start state of Gold generator
start2=[1 1 1 1 0];
start3=[1 1 1 0 0];
start4=[1 1 0 0 0];
start5=[1 0 0 0 0];
start6=[1 0 0 0 1];
start7=[1 0 0 1 0];
start8=[1 0 0 1 1];
start9=[1 0 1 0 0];
start10=[1 0 1 0 1];
%
% Generate Gold codes 1-10
%
g1=gold(N,start1);
g2=gold(N,start2);
g3=gold(N,start3);
g4=gold(N,start4);
g5=gold(N,start5);
g6=gold(N,start6);
g7=gold(N,start7);
g8=gold(N,start8);
g9=gold(N,start9);
g10=gold(N,start10);
%
for i=1:L
```

```
  xx(i)=g1(i)+g2(i)+g3(i)+g4(i)+g5(i)+g6(i)+g7(i)+g8(i)+g9(i)+g10(i);
end
%
%
% Input all 9-order peak locations for Gold codes
%
pq(1,1)=3;
pq(1,2)=5;
pq(2,1)=6;
pq(2,2)=10;
pq(3,1)=12;
pq(3,2)=20;
pq(4,1)=24;
pq(4,2)=9;
pq(5,1)=17;
pq(5,2)=18;
pq(6,1)=7;
pq(6,2)=16;
pq(7,1)=14;
pq(7,2)=1;
pq(8,1)=28;
pq(8,2)=2;
pq(9,1)=25;
pq(9,2)=4;
pq(10,1)=19;
pq(10,2)=8;
pq(11,1)=11;
pq(11,2)=23;
pq(12,1)=22;
pq(12,2)=15;
pq(13,1)=13;
pq(13,2)=30;
pq(14,1)=26;
pq(14,2)=29;
pq(15,1)=21;
pq(15,2)=27;
%
rs(1,1)=1;
rs(1,2)=12;
rs(2,1)=2;
rs(2,2)=24;
rs(3,1)=4;
rs(3,2)=17;
rs(4,1)=8;
rs(4,2)=3;
rs(5,1)=16;
rs(5,2)=6;
rs(6,1)=5;
rs(6,2)=28;
rs(7,1)=10;
rs(7,2)=25;
rs(8,1)=20;
rs(8,2)=19;
rs(9,1)=9;
rs(9,2)=7;
rs(10,1)=18;
```

```
rs(10,2)=14;
rs(11,1)=11;
rs(11,2)=30;
rs(12,1)=22;
rs(12,2)=29;
rs(13,1)=13;
rs(13,2)=27;
rs(14,1)=26;
rs(14,2)=23;
rs(15,1)=21;
rs(15,2)=15;
%
mns=0;
vss=0;
nod=0;
for iii=1:nii

ss=0;
%
for ipq=1:15
  for irs=1:15
% Set values p q r p+r q+r s p+s q+s
i1=pq(ipq,1);
i2=pq(ipq,2);
i3=rs(irs,1);
i4=i1+i3;
i5=i2+i3;
i6=rs(irs,2);
i7=i1+i6;
i8=i2+i6;
%
%   set x(32-62) etc to same values - sequence periodic - after adding noise
%
for i=1:31
  x(i+31)=xx(i)+randn*fn;
  x(i+62)=x(i+31);
  x(i+93)=x(i+31);
end
%
%   calculate 9th-O-CF c(i1,i2,i3,i4,i5,i6,i7,i8)
%
%
s=0;
for k=1:31
  y=x(k)*x(k+i1)*x(k+i2)*x(k+i3)*x(k+i4)*x(k+i5)*x(k+i6)*x(k+i7)*x(k+i8);
  s=s+y;
end
si(ipq,irs)=s/31;
ss=ss+s/31;
end
end
%
mn=ss/225;
amn(iii)=mn;
if mn > 0.9
  nod=nod+1;
```

```
else
  nod=nod+0;
end
%
mns=mns+mn;
%
%disp(si)
%
vs=0;
for ipq=1:15
  for irs=1:15
    vs=vs+(si(ipq,irs)-mn)^2;
  end
end
vs=vs/225;
vss=vss+vs;
end
%
mns=mns/10;
vss=vss/100;
%disp('MEAN  VAR')
%disp(mns)
%disp(vss)
%
nod=100*nod/nii;
disp('Detection %')
disp(nod)
%disp(amn)
disp('***************************************')
anii=nii/nav;
nod=0;
%
for ii=1:anii
  s=0;
  n=nav*(ii-1);
  for i=n+1:n+nav
    s=s+amn(i);
  end
  s=s/nav;
  bmn(ii)=s;
  if s > 0.9
    nod=nod+1;
  else
    nod=nod+0;
  end
end
%
disp('Detection % based on nav-averages')
nod=100*nod/anii;
disp(nod)
disp(bmn)
%
```

NCFGoldnonp.m

```
% NCFGoldnonp.m
% 9th-order-CF of Gold codes using combinations of TCF non-peak locations: detection %
%
fn=1
nii=1000
nav=10
%
pq(1,1)=3;
pq(1,2)=5;
pq(2,1)=6;
pq(2,2)=10;
pq(3,1)=12;
pq(3,2)=20;
pq(4,1)=24;
pq(4,2)=9;
pq(5,1)=17;
pq(5,2)=18;
pq(6,1)=7;
pq(6,2)=16;
pq(7,1)=14;
pq(7,2)=1;
pq(8,1)=28;
pq(8,2)=2;
pq(9,1)=25;
pq(9,2)=4;
pq(10,1)=19;
pq(10,2)=8;
pq(11,1)=11;
pq(11,2)=23;
pq(12,1)=22;
pq(12,2)=15;
pq(13,1)=13;
pq(13,2)=30;
pq(14,1)=26;
pq(14,2)=29;
pq(15,1)=21;
pq(15,2)=27;
%
rs(1,1)=1;
rs(1,2)=12;
rs(2,1)=2;
rs(2,2)=24;
rs(3,1)=4;
rs(3,2)=17;
rs(4,1)=8;
rs(4,2)=3;
rs(5,1)=16;
rs(5,2)=6;
rs(6,1)=5;
rs(6,2)=28;
rs(7,1)=10;
rs(7,2)=25;
rs(8,1)=20;
rs(8,2)=19;
```

```
rs(9,1)=9;
rs(9,2)=7;
rs(10,1)=18;
rs(10,2)=14;
rs(11,1)=11;
rs(11,2)=30;
rs(12,1)=22;
rs(12,2)=29;
rs(13,1)=13;
rs(13,2)=27;
rs(14,1)=26;
rs(14,2)=23;
rs(15,1)=21;
rs(15,2)=15;
%
mns=0;
vss=0;
nod=0;
for iii=1:nii

ss=0;
%
for ipq=1:15
  for irs=1:15
% Set values p q r p+r q+r s p+s q+s
i1=pq(ipq,1);
i2=pq(ipq,2)+1;
if i2>30
  i2=2;
end
i3=rs(irs,1);
i4=i1+i3;
i5=i2+i3;
i6=rs(irs,2)+1;
if i6>30
  i6=2;
end
i7=i1+i6;
i8=i2+i6;
%
x=[1 1 1 -1 -1 1 -1 -1 -1 -1 1 -1 -1 1 -1 1 1 1 -1 -1 -1 -1 -1 -1 1 -1 1 1 1 1 1];
%
%  set x(32-62) etc to same values - sequence periodic - after adding noise
%
for i=1:31
  x(i+31)=x(i)+randn*fn;
  x(i+62)=x(i+31);
  x(i+93)=x(i+31);
end
%
%  calculate 9th-O-CF c(i1,i2,i3,i4,i5,i6,i7,i8)
%
%
s=0;
for k=1:31
  y=x(k)*x(k+i1)*x(k+i2)*x(k+i3)*x(k+i4)*x(k+i5)*x(k+i6)*x(k+i7)*x(k+i8);
```

```
  s=s+y;
end
si(ipq,irs)=s/31;
ss=ss+s/31;
end
end
%
mn=ss/225;
amn(iii)=mn;
if mn > 0.9
  nod=nod+1;
else
  nod=nod+0;
end
%
mns=mns+mn;
%
%disp(si)
%
vs=0;
for ipq=1:15
  for irs=1:15
    vs=vs+(si(ipq,irs)-mn)^2;
  end
end
vs=vs/225;
vss=vss+vs;
end
%
mns=mns/10;
vss=vss/100;
%disp('MEAN  VAR')
%disp(mns)
%disp(vss)
%
nod=100*nod/nii;
disp('Detection %')
disp(nod)
%disp(amn)
disp('****************************************')
anii=nii/nav;
nod=0;
%
for ii=1:anii
  s=0;
  n=nav*(ii-1);
  for i=n+1:n+nav
    s=s+amn(i);
  end
  s=s/nav;
  bmn(ii)=s;
  if s > 0.9
    nod=nod+1;
  else
    nod=nod+0;
  end
```

```
end
%
disp('Detection % based on nav-averages')
nod=100*nod/anii;
disp(nod)
disp(bmn)
%
```

TCFGold10.m

```
% TCF of 10 Gold codes from the same u & v
%
clear all
xnfac=0;
N=5;
L=2^N-1;
start1 = [1 1 1 1 1]; % Starting state of Gold generator
start2 = [1 1 1 1 0];
start3 = [1 1 1 0 0];
start4 = [1 1 0 0 0];
start5 = [1 0 0 0 0];
start6 = [1 0 0 0 1];
start7 = [1 0 0 1 0];
start8 = [1 0 0 1 1];
start9 = [1 0 1 0 0];
start10 = [1 0 1 0 1];
%
% Generate Gold codes 1-10
%
g1=gold(N,start1);
g2=gold(N,start2);
g3=gold(N,start3);
g4=gold(N,start4);
g5=gold(N,start5);
g6=gold(N,start6);
g7=gold(N,start7);
g8=gold(N,start8);
g9=gold(N,start9);
g10=gold(N,start10);
%
for i=1:L
  x(i)=g1(i)+g2(i)+g3(i)+g4(i)+g5(i)+g6(i)+g7(i)+g8(i)+g9(i)+g10(i);
end
%  set x(32-62) to same values - sequence periodic
%
for i=1:31
  x(i+31)=x(i);
end
%
% add noise
%
for i=1:62
  x(i)=x(i)+xnfac*randn;
end
%
```

```
% calculate TCF c(i,j)
%
for i=1:30
  for j=1:30
    s=0;
    for k=1:31
      y=x(k)*x(k+i)*x(k+j);
      s=s+y;
    end
    c(i,j)=s;
  end
end
%
% Print tcf array, 10 cols at a time
%
disp('Columns 1-10')
disp(c(1:30,1:10))
disp('Columns 11-20')
disp(c(1:30,11:20))
disp('Columns 21-30')
disp(c(1:30,21:30))
figure(1);
contourf(c);
figure(2);
surf(c);
```

TCFGoldstatp.m

```
% TCFGoldstatp.m
% TCF of Gold codes with mean, var, m(4) and kurtosis for PEAK set
%
fn=1.0;
disp('Noise factor')
disp(fn)
%
% generate Gold code from preferred pair [45], [75].
%
x=[1 1 1 -1 -1 1 -1 -1 -1 -1 1 -1 -1 1 -1 1 1 1 -1 -1 -1 -1 -1 -1 1 -1 1 1 1 1 1];
%
% add Gaussian noise and repeat sequence.
%
disp('Number of iterations')
nitt=100
%
```

```
cm=0;
ca2=0;
ca4=0;
ckurt=0;
%
pq(1,1)=3;
pq(1,2)=5;
pq(2,1)=6;
pq(2,2)=10;
pq(3,1)=12;
pq(3,2)=20;
pq(4,1)=24;
pq(4,2)=9;
pq(5,1)=17;
pq(5,2)=18;
pq(6,1)=7;
pq(6,2)=16;
pq(7,1)=14;
pq(7,2)=1;
pq(8,1)=28;
pq(8,2)=2;
pq(9,1)=25;
pq(9,2)=4;
pq(10,1)=19;
pq(10,2)=8;
pq(11,1)=11;
pq(11,2)=23;
pq(12,1)=22;
pq(12,2)=15;
pq(13,1)=13;
pq(13,2)=30;
pq(14,1)=26;
pq(14,2)=29;
pq(15,1)=21;
pq(15,2)=27;
%
rs(1,1)=1;
rs(1,2)=12;
rs(2,1)=2;
rs(2,2)=24;
rs(3,1)=4;
rs(3,2)=17;
rs(4,1)=8;
rs(4,2)=3;
rs(5,1)=16;
rs(5,2)=6;
rs(6,1)=5;
rs(6,2)=28;
rs(7,1)=10;
rs(7,2)=25;
rs(8,1)=20;
rs(8,2)=19;
rs(9,1)=9;
rs(9,2)=7;
rs(10,1)=18;
rs(10,2)=14;
```

```
rs(11,1)=11;
rs(11,2)=30;
rs(12,1)=22;
rs(12,2)=29;
rs(13,1)=13;
rs(13,2)=27;
rs(14,1)=26;
rs(14,2)=23;
rs(15,1)=21;
rs(15,2)=15;
%
for nit=1:nitt
  %
  for i=1:31
    y(i)=x(i)+randn*fn;
    y(i+31)=y(i);
  end
%
% calculate TCF c(i,j)
%
for i=1:30
  for j=1:30
    s=0;
    for k=1:31
      z=y(k)*y(k+i)*y(k+j);
      s=s+z;
    end
    c(i,j)=s;
  end
end
%
% print TCF array, 10 cols at a time
%
%disp('Columns 1-10')
%disp(c(1:30,1:10))
%disp('Columns 11-20')
%disp(c(1:30,11:20))
%disp('Columns 21-30')
%disp(c(1:30,21:30))
%disp('****************************************')
%
% Calculate statistics
%
am=0;
for i=1:15
  i11=pq(i,1);
  i12=pq(i,2);
  i21=rs(i,1);
  i22=rs(i,2);
  am=am+c(i11,i12)+c(i21,i22);
end
mean=am/30;
%disp('Mean')
%disp(mean)
cm=cm+mean;
zm(nit)=mean;
```

```
%
a2=0;
a4=0;
for i=1:15
  i11=pq(i,1);
  i12=pq(i,2);
  i21=rs(i,1);
  i22=rs(i,2);
  b2=c(i11,i12)-mean;
  b2=b2*b2;
  a2=a2+b2;
  a4=a4+b2*b2;
  b2=c(i21,i22)-mean;
  b2=b2*b2;
  a2=a2+b2;
  a4=a4+b2*b2;
end
a2=a2/30;
a4=a4/30;
kurt=0;
if a2>0.1
  kurt=a4/(a2*a2);
end
%disp('Variance')
%disp(a2)
%disp('4th moment about mean')
%disp(a4)
%disp('Kurtosis')
%disp(kurt)
ca2=ca2+a2;
ca4=ca4+a4;
ckurt=ckurt+kurt;
%
z2(nit)=a2;
zk(nit)=kurt;
%
end
%
cm=cm/nitt;
ca2=ca2/nitt;
ca4=ca4/nitt;
ckurt=ckurt/nitt;
disp('OVERALL MEANS of mean, var, m(4), kurtosis')
disp(cm)
disp(ca2)
disp(ca4)
disp(ckurt)
%
disp('OVERALL VARS OF mean, var, kurtosis')
vm=0;
v2=0;
vk=0;
%
for i=1:nitt
  d=zm(i)-cm;
  vm=vm+d*d;
```

```
  d=z2(i)-ca2;
  v2=v2+d*d;
  d=zk(i)-ckurt;
  vk=vk+d*d;
end
%
vm=vm/nitt;
v2=v2/nitt;
vk=vk/nitt;
disp(vm)
disp(v2)
disp(vk)
%
```

TCFGoldstatnp.m

```
% TCFGoldstatnp.m
% TCF with mean, var, m(4) and kurtosis - for non-peak set
%
fn=1.0;
disp('Noise factor')
disp(fn)
%
% generate Gold code from preferred pair [45], [75].
%
x=[1 1 1 -1 -1 1 -1 -1 -1 -1 1 -1 -1 1 -1 1 1 1 -1 -1 -1 -1 -1 -1 1 -1 1 1 1 1 1];
%
% add Gaussian noise and repeat sequence.
%
disp('Number of iterations')
nitt=100
%
cm=0;
ca2=0;
ca4=0;
ckurt=0;
%
pq(1,1)=3;
pq(1,2)=5;
pq(2,1)=6;
pq(2,2)=10;
pq(3,1)=12;
pq(3,2)=20;
pq(4,1)=24;
pq(4,2)=9;
pq(5,1)=17;
pq(5,2)=18;
pq(6,1)=7;
pq(6,2)=16;
pq(7,1)=14;
pq(7,2)=1;
pq(8,1)=28;
pq(8,2)=2;
pq(9,1)=25;
pq(9,2)=4;
```

```
pq(10,1)=19;
pq(10,2)=8;
pq(11,1)=11;
pq(11,2)=23;
pq(12,1)=22;
pq(12,2)=15;
pq(13,1)=13;
pq(13,2)=30;
pq(14,1)=26;
pq(14,2)=29;
pq(15,1)=21;
pq(15,2)=27;
%
rs(1,1)=1;
rs(1,2)=12;
rs(2,1)=2;
rs(2,2)=24;
rs(3,1)=4;
rs(3,2)=17;
rs(4,1)=8;
rs(4,2)=3;
rs(5,1)=16;
rs(5,2)=6;
rs(6,1)=5;
rs(6,2)=28;
rs(7,1)=10;
rs(7,2)=25;
rs(8,1)=20;
rs(8,2)=19;
rs(9,1)=9;
rs(9,2)=7;
rs(10,1)=18;
rs(10,2)=14;
rs(11,1)=11;
rs(11,2)=30;
rs(12,1)=22;
rs(12,2)=29;
rs(13,1)=13;
rs(13,2)=27;
rs(14,1)=26;
rs(14,2)=23;
rs(15,1)=21;
rs(15,2)=15;
%
for nit=1:nitt
  %
  for i=1:31
    y(i)=x(i)+randn*fn;
    y(i+31)=y(i);
  end
  %
% calculate TCF c(i,j)
%
for i=1:30
  for j=1:30
    s=0;
```

```
    for k=1:31
      z=y(k)*y(k+i)*y(k+j);
      s=s+z;
    end
    c(i,j)=s;
  end
end
%
% print TCF array, 10 cols at a time
%
%disp('Columns 1-10')
%disp(c(1:30,1:10))
%disp('Columns 11-20')
%disp(c(1:30,11:20))
%disp('Columns 21-30')
%disp(c(1:30,21:30))
%disp('****************************************')
%
% Calculate statistics
%
am=0;
for i=1:15
  i11=pq(i,1);
  i12=pq(i,2)+1;
  if i12>30;
    i12=2;
  end
  i21=rs(i,1);
  i22=rs(i,2)+1;
  if i22>30
    i22=2;
  end
  am=am+c(i11,i12)+c(i21,i22);
end
mean=am/30;
%disp('Mean')
%disp(mean)
cm=cm+mean;
zm(nit)=mean;
%
% create non-peak location - replace q by q+1 or 2 if q+1>30; same for s.
a2=0;
a4=0;
for i=1:15
  i11=pq(i,1);
  i12=pq(i,2)+1;
  if i12>30
    i12=2;
  end
  i21=rs(i,1);
  i22=rs(i,2)+1;
  if i22>30
    i22=2;
  end
  b2=c(i11,i12)-mean;
  b2=b2*b2;
```

```
  a2=a2+b2;
  a4=a4+b2*b2;
  b2=c(i21,i22)-mean;
  b2=b2*b2;
  a2=a2+b2;
  a4=a4+b2*b2;
end
a2=a2/30;
a4=a4/30;
kurt=0;
if a2>0.1
  kurt=a4/(a2*a2);
end
%disp('Variance')
%disp(a2)
%disp('4th moment about mean')
%disp(a4)
%disp('Kurtosis')
%disp(kurt)
%
z2(nit)=a2;
zk(nit)=kurt;
%
ca2=ca2+a2;
ca4=ca4+a4;
ckurt=ckurt+kurt;
%
end
%
cm=cm/nitt;
ca2=ca2/nitt;
ca4=ca4/nitt;
ckurt=ckurt/nitt;
disp('OVERALL MEANS of mean, var, m(4), kurtosis')
disp(cm)
disp(ca2)
disp(ca4)
disp(ckurt)
%
disp('OVERALL VARS of mean, var, kurtosis')
%
vm=0;
v2=0;
vk=0;
%
for i=1:nitt
  d=zm(i)-cm;
  vm=vm+d*d;
  d=z2(i)-ca2;
  v2=v2+d*d;
  d=zk(i)-ckurt;
  vk=vk+d*d;
end
%
vm=vm/nitt;
v2=v2/nitt;
```

```
vk=vk/nitt;
%
disp(vm)
disp(v2)
disp(vk)
%
```

Goldetmods.m

```
% 'Goldetmods'
% Gold's 2-stage method to detect Gold codes, with spectral test
% Program produces spectrum plot of detector's output, which approximates
% the data signal if the delay pairs are correct for the Gold code.
%
% specify delay pairs (p,q) and (r,s)
%
p=17;
q=18;
r=19;
s=20;
%
% Input noise factor fn and number of 31-length Gold code cycles nc
%
disp('Noise factor')
fn=1.0
nc=40;
%
x=[1 1 1 -1 -1 1 -1 -1 -1 -1 1 -1 -1 1 -1 1 1 1 -1 -1 -1 -1 -1 -1 1 -1 1 1 1 1 1];
%
% Generate nc cycles of b-modulated 31-length Gold code in y
%
for i=1:nc
  b=-1;
  xr=rand;
  if xr > 0.5
    b=1;
  end
  bb(i)=b;
  ia=(i-1)*31;
  for j=1:31
    y(ia+j)=b*x(j)+randn*fn;
  end
end
% Calculate triple product for shift pair (p,q)
%
ni=nc*31-30;
for i=1:ni
  z(i)=y(i)*y(i+p)*y(i+q);
end
%
% Calculate triple product of 1st stage output for shift pair (r,s)
%
ni=ni-30;
for i=1:ni
  t(i)=z(i)*z(i+r)*z(i+s);
```

```
end
%
%for i=1:124
%   c(i)=t(i);
%   bbb(i)=bb(i);
%end
%disp(c)
%disp('*************************************')
%disp(bbb)
%disp('*************************************')
%
% Calculate mean and variance of 2nd stage output
%
s=0;
for i=1:ni
  s=s+t(i);
end
s=s/ni;
%
ss=0;
for i=1:ni
  d=t(i)-s;
  ss=ss+(d*d);
end
ss=ss/ni;
%
disp('No. of output values averaged')
disp(ni)
disp('MEAN and VARIANCE')
disp(s)
disp(ss)
%
fy=fft(t,1024);
Py=fy.*conj(fy)/1024;
%
for i=1:128
  j=(i-1)*4;
  cs=0;
  for k=1:4
    cs=cs+Py(j+k);
  end
  Ps(i)=cs/4;
end
%
f=1000*(0:127)/1024;
plot(f,Ps(1:128))
title('Spectrum of output for correct shift-pairs. 0 dB noise level');
%
```

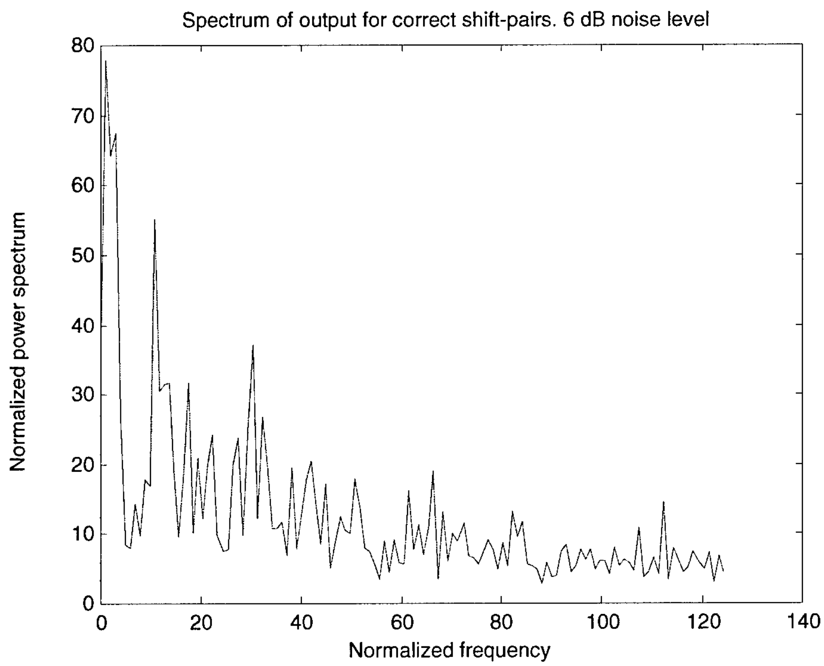
Goldetmodsn.m

```
% 'Goldetmodsn'
% Gold's 2-stage method to detect Gold codes, with spectrum plot
%
% specify delay pairs (p,q) and (r,s) - non-peaks
```

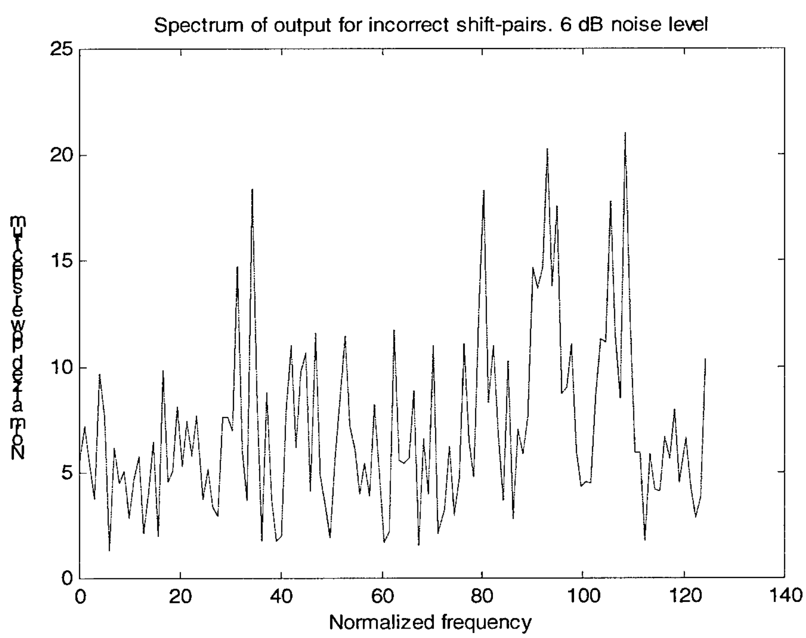
```
%
p=17;
q=19;
r=20;
s=21;
%
% Input noise factor fn and number of 31-length Gold code cycles nc
%
disp('Noise factor')
fn=1.0
nc=40;
%
x=[1 1 1 -1 -1 1 -1 -1 -1 -1 1 -1 -1 1 -1 1 1 1 -1 -1 -1 -1 -1 -1 1 -1 1 1 1 1 1];
%
% Generate nc cycles of b-modulated 31-length Gold code in y
%
for i=1:nc
  b=-1;
  xr=rand;
  if xr > 0.5
    b=1;
  end
  bb(i)=b;
  ia=(i-1)*31;
  for j=1:31
    y(ia+j)=b*x(j)+randn*fn;
  end
end
% Calculate triple product for shift pair (p,q)
%
ni=nc*31-30;
for i=1:ni
  z(i)=y(i)*y(i+p)*y(i+q);
end
%
% Calculate triple product of 1st stage output for shift pair (r,s)
%
ni=ni-30;
for i=1:ni
  t(i)=z(i)*z(i+r)*z(i+s);
end
%
%for i=1:124
%   c(i)=t(i);
%   bbb(i)=bb(i);
%end
%disp(c)
%disp('************************************')
%disp(bbb)
%disp('************************************')
%
% Calculate mean and variance of 2nd stage output
%
s=0;
for i=1:ni
  s=s+t(i);
```

```
end
s=s/ni;
%
ss=0;
for i=1:ni
  d=t(i)-s;
  ss=ss+(d*d);
end
ss=ss/ni;
%
disp('No. of output values averaged')
disp(ni)
disp('MEAN and VARIANCE')
disp(s)
disp(ss)
%
fy=fft(t,1024);
Py=fy.*conj(fy)/1024;
%
for i=1:128
  j=(i-1)*4;
  cs=0;
  for k=1:4
    cs=cs+Py(j+k);
  end
  Ps(i)=cs/4;
end
%
f=1000*(0:127)/1024;
plot(f,Ps(1:128))
%
```

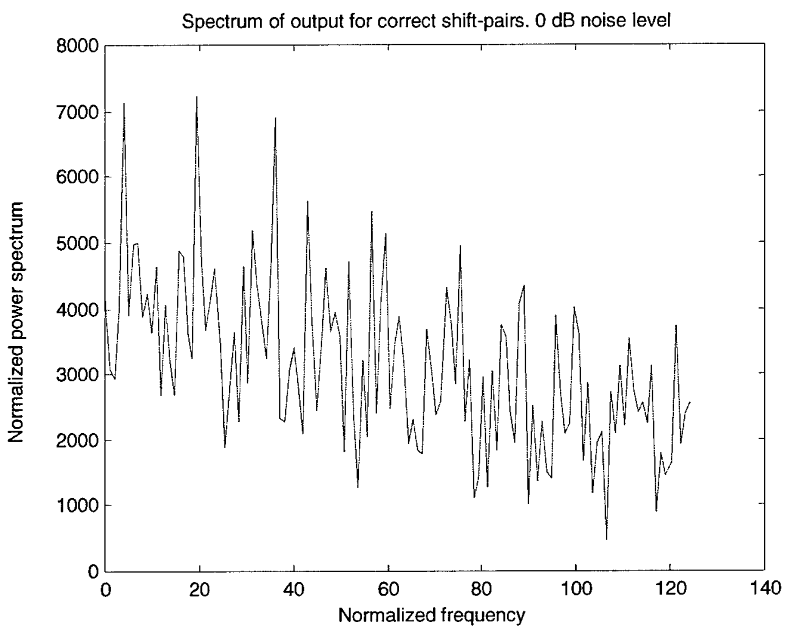
Graph produced by Matlab routine *Goldetmods.m.*



Graph produced by Matlab routine *Goldetmodsn.m.*



Graph produced by Matlab routine *Goldetmods.m.*



Graph produced by Matlab routine *Goldetmodsn.m.*